NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ATTRACTOR BASINS OF VARIOUS ROOT-FINDING METHODS

by

Bart D. Stewart

June 2001

Thesis Advisor:	David Canright
Second Reader:	Carlos F. Borges

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Title (Mix case letters) **ATTRACTOR BASINS OF VARIOUS ROOT-FINDING METHODS**	**5. FUNDING NUMBERS**
6. AUTHOR(S) Stewart, Bart D.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

Real world phenomena commonly exhibit nonlinear relationships, complex geometry, and intricate processes. Analytic or exact solution methods only address a minor class of such phenomena. Consequently, numerical approximation methods, such as root-finding methods, can be used.

The goal is, by making use of a variety of root-finding methods (Newton-Rhapson, Chebyshev, Halley and Laguerre), to gain a qualitative appreciation on how various root-finding methods address many prevailing real-world concerns, to include, how are suitable approximation methods determined; when do root finding methods converge; and how long for convergence?

Answers to the questions were gained through examining the basins of attraction of the root-finding methods. Different methods generate different basins of attraction. In the end, each method appears to have its own advantages and disadvantages.

14. SUBJECT TERMS Basin of Attraction, Numerical Methods, Complex Polynomials			**15. NUMBER OF PAGES** 120
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

ATTRACTOR BASINS OF VARIOUS ROOT-FINDING METHODS

Bart D. Stewart
Captain, United States Army
B.S., United States Military Academy, 1993
M.S., Troy State University, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2001

Author: ______________________________
Bart D. Stewart

Approved by: David Canright
David Canright, Thesis Advisor

Carlos F. Borges
Carlos F. Borges, Second Reader

M. A. Morgan
Michael A. Morgan, Chairman
Department of Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Real world phenomena commonly exhibit nonlinear relationships, complex geometry, and intricate processes. Analytic or exact solution methods only address a minor class of such phenomena. Consequently, numerical approximation methods, such as root-finding methods, can be used.

The goal is, by making use of a variety of root-finding methods (Newton-Rhapson, Chebyshev, Halley and Laguerre), to gain a qualitative appreciation on how various root-finding methods address many prevailing real-world concerns, to include, how are suitable approximation methods determined; when do root finding methods converge; and how long for convergence?

Answers to the questions were gained through examining the basins of attraction of the root-finding methods. Different methods generate different basins of attraction. In the end, each method appears to have its own advantages and disadvantages.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

ACKNOWLEDGMENTS

The author wishes to wholeheartedly thank Professor David Canright for his structured guidance, absolute patience, and genuine enthusiasm throughout the thesis effort. Without him, many wrong paths would have been traveled.

Additionally, the author extends thanks to Professor Beny Neta for a very thorough explanation of Popovski's One-Point Approximation Method. Again to Professor Neta and Professor Carlos F. Borges for always keeping an open door policy.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Root finding methods have been of interest for a long time. Why? Often people ask qualitative questions about real-world phenomena, and they want these questions answered. To come to an answer, one must accurately model the real world phenomena in a mathematical model, and then solve the model. In many applications, the solution involves finding a root.

Constructing models is rarely a simple process. Models come in many shapes and sizes. Some of these represent a dynamical process – a recipe for how real-world phenomena interact and change over time. How these interactions and changes occur governs the choice of model. For example, the continuous model leading to a differential equation is reasonable for certain phenomena, while difference equations in the form of a recurrence relation address phenomena occurring in discrete steps. Solutions, however, are not guaranteed in every instance.

When analytical or exact methods are applicable, sometimes formulas for solutions exist. However, these methods are restrictive, often providing insight into the behavior of only a minor class of real world phenomena. Included in this category are models that can be approximated by linear relationships, simple geometry, and low dimensionality. For a great deal of real world phenomena, that is not the norm. Real world phenomena commonly exhibit nonlinear relationships, complex geometry, and intricate processes. Consequently, exact methods can be of limited practical value (Chapra, 1988).

B. MOTIVATION

Where analytical or exact methods fail, numerical approximation methods often succeed, approximately. One such approximation method employs difference equations. When applied to a large though finite number of steps, difference equations are closely related to the continuous behavior of a differential equation (Figure 1.1) In fact, a continuous model, *y(t)*, can be seen as a limit of the discrete model, $y_n(t_n)$ (Figure 1.2).

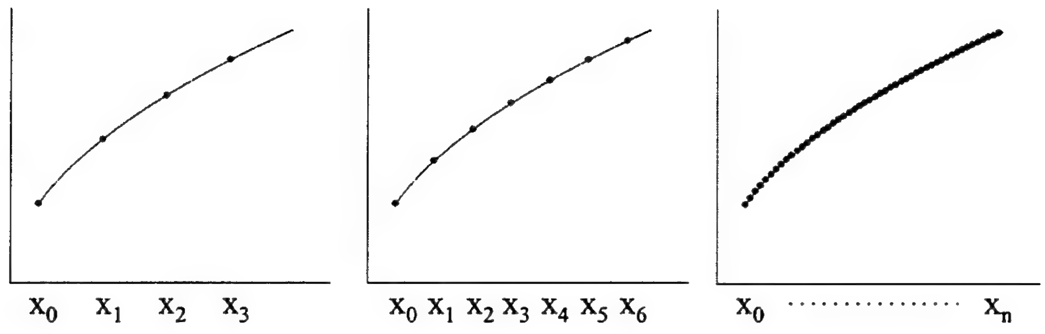


Figure 1.1. Approximating Continuous Behavior

$$\begin{aligned} \frac{\partial y}{\partial t} &= f(t,y) \\ &= \lim_{\Delta t \to 0} \frac{y_{n+1} - y_n}{\Delta t} \\ &\approx \frac{y_{n+1} - y_n}{\Delta t} \\ f(t_n, y_n) &\approx \frac{y_{n+1} - y_n}{\Delta t} \\ y_{n+1} &\approx y_n + \Delta t[f(t_n, y_n)] \end{aligned}$$

Figure 1.2. Approximating a Differential Equation Using a Difference Equation

Although the model approaches are different, solution methods for each share common ground. In the continuous model, solution curves may be obtained from the roots of a linear, constant-coefficient differential equation's characteristic polynomial. In the discrete model, solutions come from the roots of the recurrence relation's characteristic polynomial. In either case, roots can be real, imaginary, or complex. Consequently, solutions can vary greatly in their dynamical behaviors.

Numerical (Root finding) methods, however, serve as the computational tools that unveil the mysteries of such dynamical behavior. Different methods, however, may produce different results from the same initial guess. So things can get really interesting!

C. GOALS

This thesis seeks to gain a qualitative appreciation on how various root-finding methods address many prevailing real-world concerns, to include, how are suitable approximation methods determined; when do root finding methods converge; and how long for convergence? Particular emphasis is given to finding which initial guesses lead to which roots.

D. METHODOLGY

From a mesh of points within the complex plane, Newton-Raphson, Chebyshev, Halley, and Laguerre root-finding methods numerically compute the successive approximations of some n^{th} order complex polynomial's roots. In order to better grasp the effects, the results are mapped – thus, creating a geometry of basins of attractions which are the set of starting points whose trajectories are asymptotic to a bounded region (Devaney, 1989). The geometrical differences lend to the qualitative difference amongst the root-finding methods.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BEHAVIORS OF DYNAMICAL SYSTEMS

While the mathematics describing dynamic behavior may be fairly straightforward, interpreting such behavior can be difficult. In order to truly grasp it, one must familiarize oneself with the role of numerical methods and the utility of mapping their geometry.

A. NUMERICAL METHODS ROLE

Numerical methods approximate solutions to mathematically expressed models. When these solutions are obtained from the zeros of some functions, root-finding methods serve as the tool of choice. These methods are usually iterative – beginning with an initial starting value and computing successive approximate solutions using a well-defined recurrence relation (Figure 2.1). Each successive step yields a numerical solution to the recurrence relation – in essence, generating a sequence of even better approximations. Hence, the solution process itself is a discrete dynamical system that generates a sequence of numbers. As Table 2.2 illustrates, each term of the sequence not

Sample Recurrence Relation	Idea of Successive Approximations
$x_{n+1} = f(x_n)$	$x_n = \underbrace{f\ (\ f\ (\cdots\ f(x_0)\ \cdots\ )}_{n\ times}$

Figure 2.1. Recurrence Relation & Iteration

Sequence	*Numerical Solutions Per Iterative Step*	*Long Term Behavior*
I	$\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \cdots \rightarrow$	0 (Convergence)
II	$\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \frac{4}{2}, \frac{5}{2}, \frac{6}{2}, \frac{7}{2}, \cdots \rightarrow$	∞ (Divergence)
II	$\frac{1}{2}, \frac{2}{3}, \frac{3}{2}, \frac{9}{2}, \frac{7}{2}, \frac{9}{2}, \frac{1}{2}, \cdots \rightarrow$	?

Figure 2.2. Arbitrary Numerical Solutions

only signifies a numerical solution for the n^{th} iterative step, but also suggests the ultimate behavior of that solution. Determining such behavior is not always done by simple inspection. Some sequences are obvious; others are not. Consequently, numbers alone are often not enough.

B. UTILITY OF MAPPING – SINGLE FIXED POINT

Another, often preferred, method used to determine dynamical behaviors is to visualize them. Visualization entails mapping out the geometry of the numerical solutions. Why is this geometry important? Simply put, it graphically depicts the dynamical behavior of root-finding methods.

As Figure 2.3 suggests, the mapping of a sequence of numerical solutions depicts the behavioral path or trajectory of a single starting point. Starting points that do not change after iteration are called fixed, and qualitative behaviors of other starting points can be interpreted in relation to the fixed points.

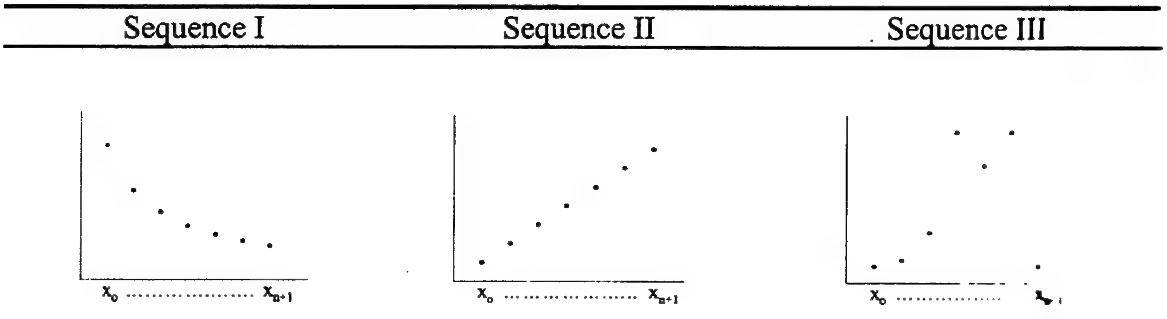


Figure 2.3. Mapping of Numerical Solutions

Cobweb diagrams help point out the qualitative behaviors near fixed points using the principle of feedback (Figure 2.4) (After Peitgen, 1992). The principle of feedback is simple – an input, x_n, is given, processed through some function, f, and then the output, y_n, becomes the next input, x_{n+1}, repeatedly. When allowing the ouput to equal the next input, an identity exists so that $(x_{n+1} = y_n) \equiv (x = y)$. Cobweb diagrams exploit the relationship, map the iterations, and reveal the behaviors of fixed points.

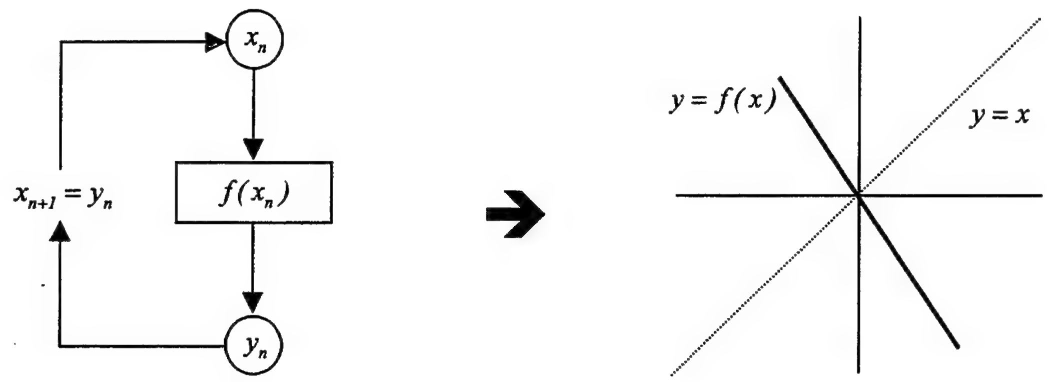


Figure 2.4. Cobweb Diagram from the Principle of Feedback

Behaviors about fixed points are converging, diverging or chaotic, and all can be mapped. Convergent mappings point out attracting fixed points; divergent mappings denote repulsive ones; and chaotic mappings never settle. While all three behaviors are essential in describing dynamical behavior, a simple example excluding chaotic mappings will suffice to illustrate the point. Consider a simple linear recurrence relation; $x_{n+1} = f(x_n) = mx_n + b$. When $|f'(x_n)| < 1$, the mapping contracts, converging to a fixed point. When $|f'(x_n)| > 1$, the mapping expands diverging off to positive or negative infinity. Figure 2.5 clarifies the point.

With relatively little effort, the geometrical approach can handle nonlinear behavior as well. For smooth nonlinear recurrence relations, the same sort of contracting and expanding argument holds near the fixed point. As Figure 2.6 indicates, the trick is to locally reduce the nonlinear model to linear parts, apply the graphical analysis, and then couple the pieces together.

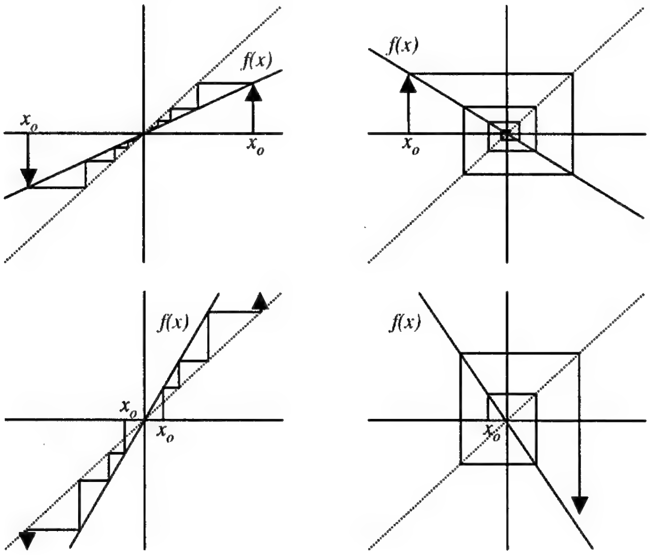


Figure 2.5. Cobweb Diagrams

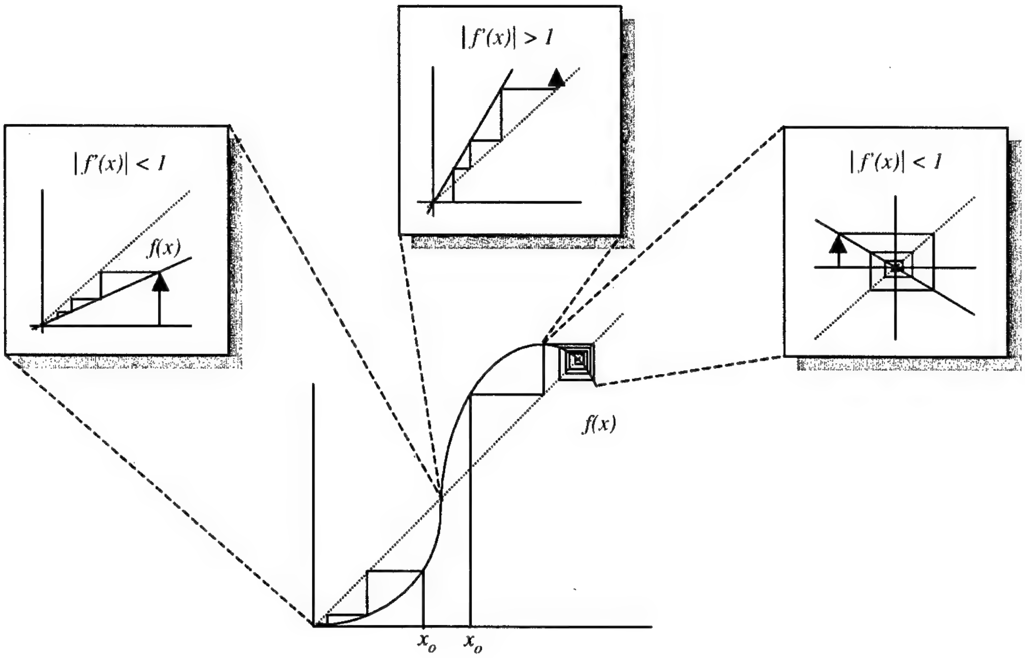


Figure 2.6. Concept of Linearizing a Nonlinear Mapping Function

Although dynamic behaviors about a single fixed point are fairly predictable, startling behavioral effects can and often do occur when multiple fixed points exist.

C. UTILITY OF MAPPING – MULTIPLE FIXED POINT

When fixed points coexist, the geometry of numerical solutions can change considerably. The effect of each fixed point is no longer simple; rather their effects interact. Consequently, determining such points and their effect is a necessity.

Multiple fixed points are often found in the realm of nonlinear phenomena. While the effect of a single fixed point has been discussed, what happens when there are two, three, or *n* of them? How many can there be when the iterator is a root approximation method? As Figure 2.7 points out, the fundamental theorem of algebra tells us that an n^{th} degree polynomial is factorable into *n* linear factors and contains exactly *n* roots, which are not necessarily distinct. Whether these points are real, imaginary or complex, their coexistence may create surprising behaviors.

Every n^{th}-order polynomial possesses exactly n roots

Any polynomial of the form

$$\begin{aligned} p(x) &= a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_2 x^2 + a_1 x + a_0 \\ &= \sum_{i=0}^{n} a_i x^i \end{aligned}$$

can always be expressed as

$$\begin{aligned} p(x) &= a_n (x - z_n)(x - z_{n-1})(x - z_{n-2}) \ldots (x - z_2)(x - z_1) \\ &= a_n \prod_{i=1}^{n} (x - z_i) \end{aligned}$$

where the points z_i are the polynomial roots, and they may be real, imaginary or complex.

Figure 2.7. Fundamental Theorem of Algebra (After Smith, 1977)

With each additional fixed point, coexisting attractors can exhibit varying behaviors. Such behaviors are actually emerging in a sort of competitive state – with each vying to influence a solution's trajectory. As Figure 2.8 suggests, such behavioral effects may or may not extend globally. Each attracting region is called a basin of attraction – the set of starting points whose trajectories are asymptotic to a bounded region. Competition amongst the fixed points, in the effect upon x_n, exists near and on basin boundaries. Moving the fixed points can create new basins and destroy old ones.

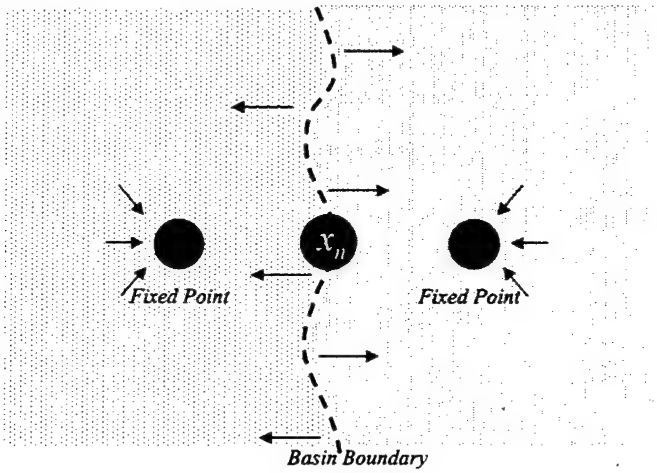


Figure 2.8. Competing Effects of Multiple Fixed Points

Basin boundaries can take on infinitely many shapes. And basin boundaries can be far more complicated than a simple curve, and in most instances are. Within their intricate patterns, commonly referred to as Julia sets, is the key that reveals some erratic behaviors. Where the basins interact and compete, the behavior is not so obvious. Figure 2.9 demonstrates how nearby starting points, which are expected to have similar behaviors, can assume distinct solution paths, particularly near basin boundaries. Consider each starting point, x_1, x_2, and x_3. Despite their 'nearness', starting points x_1 and x_3 reach a root (through different roots) while x_2, which begins on a basin boundary, never settles. Hence, behaviors have a sensitive dependence on starting points, and can, at times, be considered chaotic.

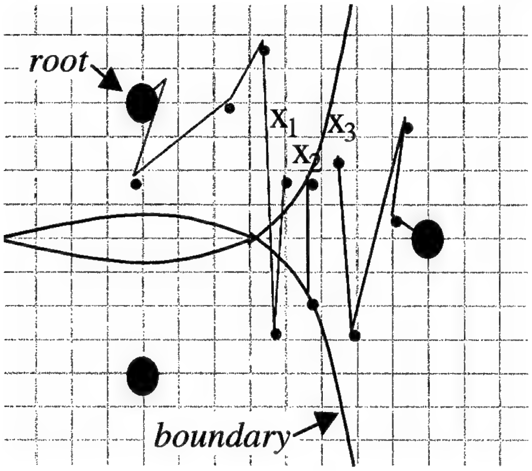


Figure 2.9. Behavioral Effects of Multiple Fixed Points (After Pergler, 1999)

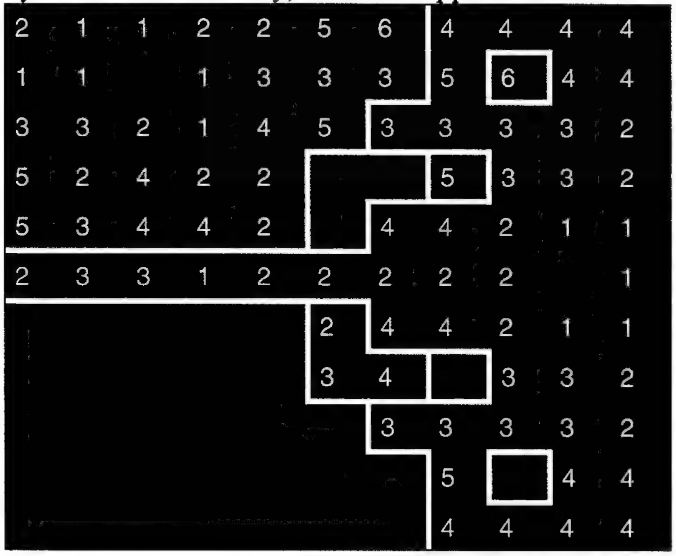
D. JULIA SETS

Further consideration of such behaviors begins with observing the role of Julia sets. Julia sets are the boundary of basins of attraction -- distinguishing which starting points are 'prisoners' to some fixed points' basin, and others that 'escape' them. Consider the following example in Figure 2.10 (After Peitgen, 1992). Note that 'prisoner' points converge to some basin, while 'escapees', had any existed, would never settle. While the Julia set may be quite complicated, its role remains crucial in revealing the coexistence and competition of complex behavior.

Rules: Within the bounded region, select an arbitrary starting point. Move about to the next point as indicated (by Newton-Rhapson's method for cube roots of unity), and continue until one, the path halts – in which the next destination is itself (marked by X), or two, the path becomes cyclical. Evaluate each starting point.

L	K2	K3	K3	K4	K4	I5	I6	I7	I8	I8	I9
K	K3	K3	☒	K3	I3	H4	H5	G7	H8	H9	H9
I	I3	I3	K4	K3	I2	G3	F5	F7	F8	G9	G10
H	H3	I4	K5	L3	K1	D2	B5	D7	F9	F9	G10
G	G4	H5	K7	K6	LA	A2	C7	C10	E10	F10	F10
F	F4	F6	F9	F10	F11	F11	F11	F11	F11	☒	F10
E	E4	D5	B7	B6	A2	L2	I7	I10	G10	F10	F10
D	D3	C4	B5	A3	B1	H2	K5	H7	F9	F9	E10
C	C3	C3	B4	B3	C2	E3	F5	F7	F8	E9	E10
B	B3	B3	☒	B3	C3	D4	D5	E7	D8	D9	D9
A	B2	B3	B3	B4	B4	C5	C6	C7	C8	C8	C9
	1	2	3	4	5	6	7	8	9	10	11

While it is apparent that three basins of attraction exist, there is another valuable piece of information. Through adding the number of moves necessary from an arbitrary starting point to a fixed point and coloring the basins of attraction, the Julia set (approximated by the white boundary) becomes apparent.



2	1	1	2	2	5	6	4	4	4	4
1	1		1	3	3	3	5	6	4	4
3	3	2	1	4	5	3	3	3	3	2
5	2	4	2	2			5	3	3	2
5	3	4	4	2		4	4	2	1	1
2	3	3	1	2	2	2	2	2		1
					2	4	4	2	1	1
					3	4		3	3	2
						3	3	3	3	2
							5		4	4
							4	4	4	4

Figure 2.10. Julia Set Example

III. NUMERICAL METHODS

Since numerical methods are capable of approximating the zeros of an analytic function, root-finding methods serve as the tool of choice. Such methods come in many shapes and sizes. Some are rather simple; others are complex. Each, however, employs a different iterative approach that affects the geometry of numerical solutions, and thus impacts on dynamical interpretations. Consequently, an investigation of such methods is necessary.

Although there are many root-finding methods to choose from, their conceptual origin is the same – that is, all stem from a successive point-wise approximation of an arbitrary function's root. For example, Taylor's theorem approximates a function value and, when truncated accordingly, is a numerical method of some sort. Figure 3.1 illustrates another, and more recent, method yielding a one parameter family, *e*, of single-point numerical methods capable of finding roots (After Popovski, 1979). Of particular interest were the Newton-Raphson, Chebyshev, Halley, and Laguerre methods.

For illustrative purposes, a uniform approach is applied. No matter the root-finding method, each considers the function, $f(z)=z^3-1$, and is restricted to real arithmetic for its geometrical interpretation. From an arbitrary point, x_o, approximations are computed so as to satisfy Popovski's single-point method, $y(z)=p_1+p_2(x-p_3)^e$, save our final method. Successive computations yield more approximations, $x_0, x_1, x_2, \ldots, x_n$. To ensure the dynamic behavior is clear, approximations are represented numerically and geometrically.

Solving $f(z)=0$ can be found on the basis of a single-point approximation by the four parameter function

$$y(z) = p_1 + p_2(z - p_3)^e$$

Consider a function $f(z)$ on an interval $[a,b]$ where $f(a)f(b)<0$ and $f'(z)f''(z)\neq 0$, $z\in[a,b]$. Let $z_i\in[a,b]$ be the i^{th} approximation to the root $r\in[a,b]$ of $f(z)=0$. Then the following approximation to the root r, z_{i+1} may be obtained from the system of equations where

$$y(z_{n+1}) = 0$$
$$y^{(d)}(z_n) = f^{(d)}(z_n) \equiv f^{(d)},\ d = 0,\ 1,\ 2.$$

and when solved yields

$$z_{n+1} = z_n + (e-1)\frac{f'(z_n)}{f''(z_n)}\left[\left(1-\frac{e}{e-1}\frac{f(z_n)f''(z_n)}{f'(z_n)^2}\right)^{1/e}-1\right]$$

Figure 3.1. Popovski's Single-Point Iteration Formula

No matter the function to be approximated, special parameter values, e, reveal familiar methods. When e approaches one, the single point iteration formula reduces to the popular Newton-Raphson method (Figure 3.2). The approximation method simply computes a tangent line to the point x_n of our function. When e equals one-half, the single point iteration formula reduces to Chebyshev's method (Figure 3.3). The approximation method, rather than line, computes a tangent horizontal parabola to the point x_n of our function. When e equals negative one, the single point iteration formula reduces to Halley's method (Figure 3.4). The approximation method computes a tangent hyperbola to the point x_n of our function. Laguerre's method takes a different approach (Figure 3.5)(Press, 1988). Rather than computing a tangent near the point x_n, the method mimics the function's behavior there – that is, an n^{th} order function receives an n^{th} order polynomial approximation. For each of these approximations, the root of its tangent or n^{th} approximation typically represents a better approximation to our function's root.

Newton-Rhapson Single Point Approximation

$$y(x) = p_1 + p_2(x - p_3)$$

$$\textit{where} \quad p_1 = f(x_n) - f'(x_n)(x_n)$$
$$p_2 = f'(x_n)$$
$$p_3 = 0$$

$$\textit{yielding} \quad y(x) = f(x_n) + f'(x_n)(x - x_n)$$

y(x) approximations...

n	x_n
0	3.5000
1	2.3605
2	1.6335
3	1.2139
4	1.0355
5	1.0012
6	1.0000

Newton-Rhapson Iterator is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

for $f(x) = x^3 - 1$...

$$x_{n+1} = x_n - \frac{x_n^3 - 1}{3x_n^2}$$

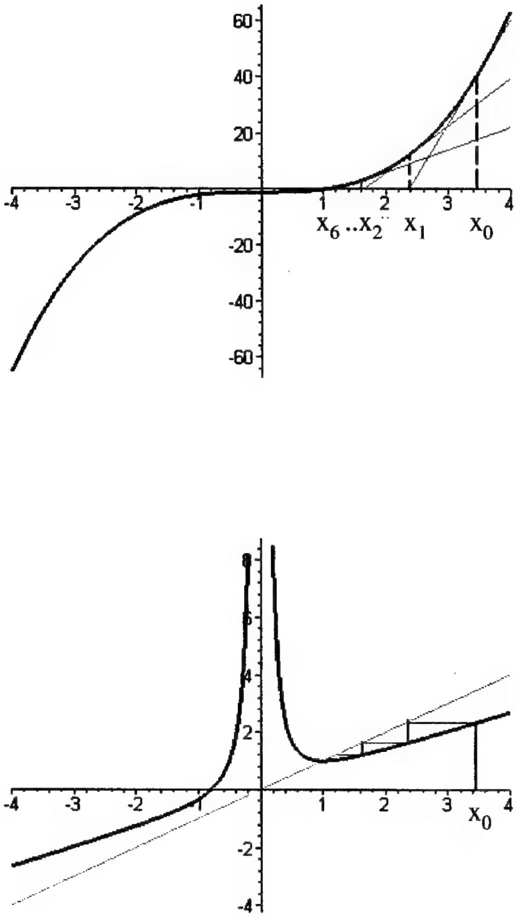


Figure 3.2. Newton-Rhapson Method

Chebyshev Single Point Approximation

$$y(x) = p_1 + p_2(x - p_3)^{1/2}$$

$$\textit{where} \qquad p_1 = f(x_n) + \frac{[f'(x_n)]^2}{f''(x_n)}$$

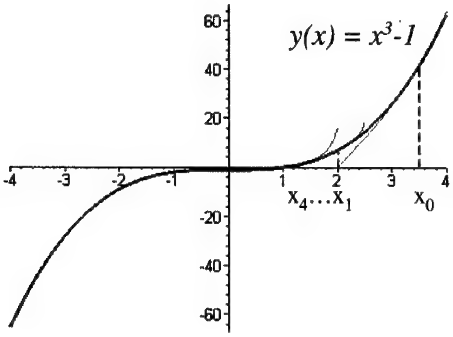
$$p_2 = 2f'(x_n)(x_n - p_3)^{1/2}$$

$$p_3 = x_n + \frac{f'(x_n)}{2f''(x_n)}$$

$$\textit{yielding} \quad y(x) = f(x_n) + \frac{f'(x_n)^2}{f''(x_n)} + 2f'(x_n)\left[-\frac{f'(x_n)}{2f''(x_n)}\right]^{1/2}\left[x - x_n - \frac{f'(x_n)}{2f''(x_n)}\right]^{1/2}$$

y(x) approximations...

n	x_n
0	3.5000
1	1.9899
2	1.2421
3	1.0125
4	1.0000



Chebyshev Iterator is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)^2 f''(x_n)}{2f'(x_n)^3}$$

for $f(x) = x^3 - 1...$

$$x_{n+1} = x_n - \frac{x_n^3 - 1}{3x_n^2} - \frac{(x_n^3-1)^2}{9x_n^5}$$

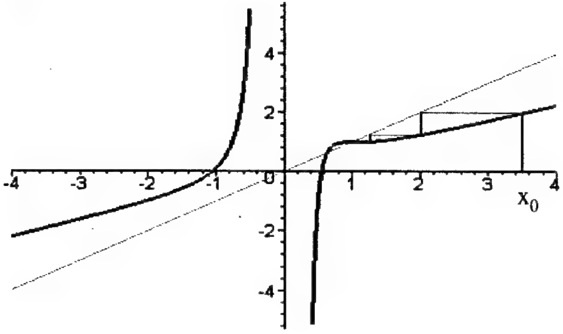


Figure 3.3. Chebyshev Method

Halley Single Point Approximation

$$y(x) = p_1 + \frac{p_2}{(x-p_3)}$$

where $$p_1 = f(x_n) - \frac{2[f'(x_n)]^2}{f''(x_n)}$$

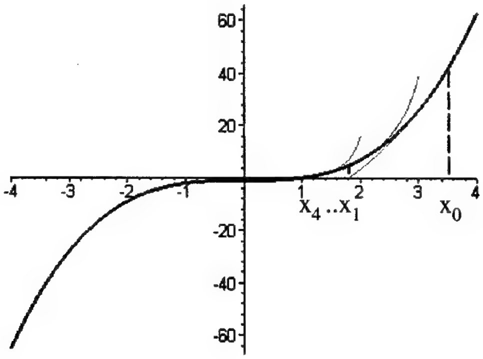
$$p_2 = -\frac{4[f'(x_n)]^3}{[f''(x_n)]^2}$$

$$p_3 = x_n + \frac{2f'(x_n)}{f''(x_n)}$$

yielding $$y(x) = f(x_n) - \frac{2f'(x_n)^2}{f''(x_n)} - \frac{4f'(x_n)^3}{f''(x_n)^2\left[x - x_n - \frac{2f'(x_n)}{f''(x_n)}\right]}$$

y(x) approximations...

n	x_n
0	3.5000
1	1.8105
2	1.1163
3	1.0009
4	1.0000



Halley Iterator is

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f(x_n)f''(x_n)}$$

for $f(x) = x^3 - 1$*...*

$$x_{n+1} = x_n - \frac{6(x_n^{\,3}-1)x_n^{\,2}}{18\,x_n^{\,4} - 6(x_n^{\,3}-1)x_n}$$

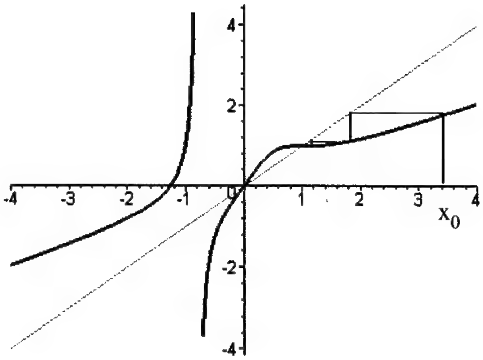


Figure 3.4. Halley Method

Examining relations between the polynomial, its roots, and its derivatives

Factor the polynomial:

$$P_n(x) = (x-x_1)(x-x_2)...(x-x_n)$$

$$\ln|P_n(x)| = \ln|(x-x_1)| + \ln|(x-x_2)| + ... + \ln|(x-x_n)|$$

Obtain derivative relations:

$$\frac{d|P_n(x)|}{dx} = \frac{1}{x-x_1} + \frac{1}{x-x_2} + ... + \frac{1}{x-x_n} = \frac{P_n'}{P_n} = G$$

$$-\frac{d^2|P_n(x)|}{dx^2} = \frac{1}{(x-x_1)^2} + \frac{1}{(x-x_2)^2} + ... + \frac{1}{(x-x_n)^2} = \left[\frac{P_n'}{P_n}\right]^2 - \frac{P''}{P_n} = H$$

Assume root x_1 is distance a from current guess, while all other roots are assumed to be located at a distance b, where

$$a = x - x_1 \qquad b = x - x_i, \quad i = 2,3,...,n$$

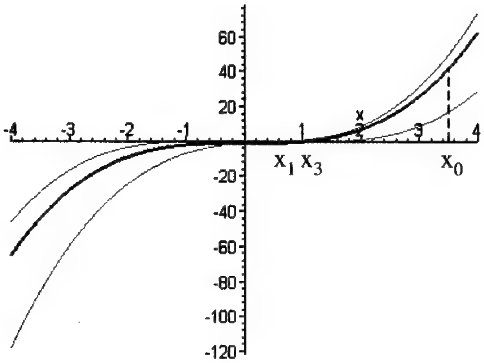
$$G = \frac{1}{a} + \frac{n-1}{b} \qquad H = \frac{1}{a^2} + \frac{n-1}{b^2}$$

$$a = \frac{n}{G \pm \sqrt{(n-1)(nH-G^2)}} \qquad b = \frac{(n-1)a}{Ga-a}$$

yielding $y(x) = [x-(x_n-a_n)]\ [x-(x_n-b_n)]^{n-1}$

y(x) approximations...

n	x_n
0	3.5000
1	0.8814
2	1.0651
3	1.0000



Laguerre Iterator is

$$x_{n+1} = x_n - \frac{n}{max\left[\frac{f'(x_n)}{f(x_n)} \pm n\left[\left(\frac{f'(x_n)}{f(x_n)}\right)^2 - \left(\frac{f''(x_n)}{f(x_n)}\right)^2\right] - \left(\frac{f'(x_n)}{f(x_n)}\right)^2 \sqrt{n-1}\right]}$$

Figure 3.5. Laguerre Method

While each method can determine the appropriate root, certain methods are preferred. As Figure 3.6 illustrates, when monotonic behavior exists, the preferential order is clear – Laguerre, Halley, Chebyshev, and then Newton-Rhapson. In Laguerre, Halley and Chebyshev, the approximating curves echo the shape of our function. Newton-Rhapson's approximating curve is restricted to a simple line. Although convergence is guaranteed, it varies according to the step sizes of the approximating methods. With the smallest step size, Newton-Rhapson is only quadratically convergent while the other methods having larger step sizes are cubically convergent. When a function is not monotonic, the preference is generally uncertain – with no single method consistently better than the others. Clues to determining such an ordering begins with the careful observation of each method's geometry.

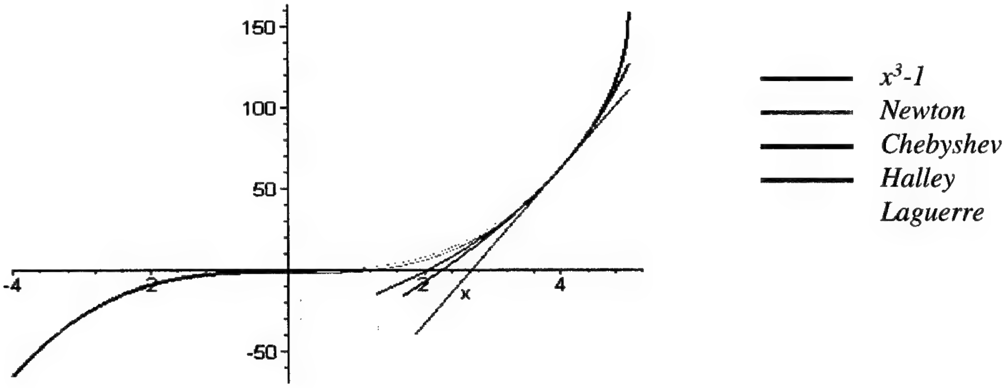


Figure 3.6. Monotonic Behavior & The Methods

THIS PAGE INTENTIONALLY LEFT BLANK

IV. NUMERICAL METHODS' GEOMETRY

Again, numerical methods can sort through a dynamical system's behavior through finding its roots and examining their affect. With different methods yielding different behaviors, an examination of individual numerical methods is necessary. Recall how our four numerical methods, while obtaining the appropriate root, all sought distinct solution paths to it. Consider now a mesh of complex starting points, rather than a single real point, with an assortment of fixed points. What are the numerical solution paths now? What is the effect of the competition and coexistence of fixed points? What numerical method is preferred, if any? Answers to these questions appear when mapping and coloring each numerical method's solutions.

While an n^{th} order complex polynomial with distinct roots partitions the complex plane into n number of basins, the partitions may or may not be equally distributed – or even connected for that matter. In an ideal setting, these attracting regions resemble a Voronoi diagram – regions containing all points that are the nearest neighbors to the polynomial's zero (Figure 4.1). Few things, though, are ideal. Rather, an attracting region contains all starting points that asymptotically approach the zero, despite their locality.

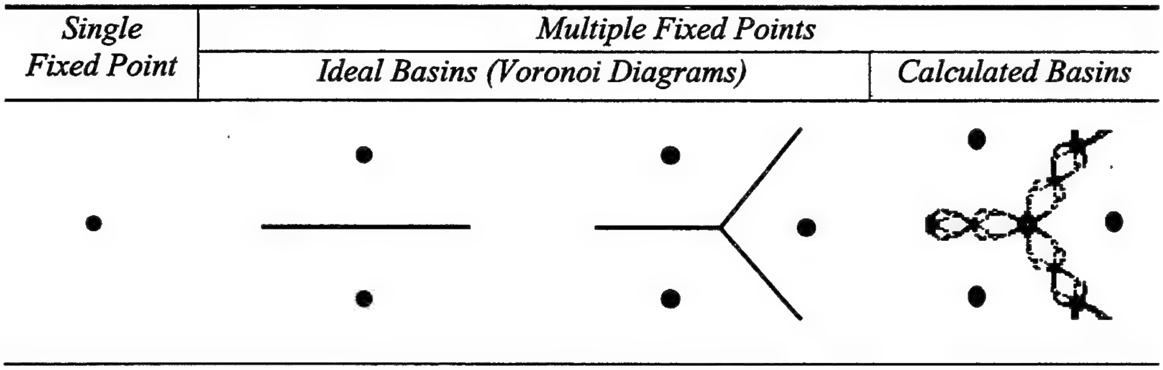


Figure 4.1. Basins of Attraction

One popular method to visualize these regions is basin coloring. The process simply assigns *n* colors to the *n* basins, executes some numerical method to calculate which initial points within a bounded region or mesh converge to a particular basin, and paints that basin's color to that point (Figure 4.2). Furthermore, the number of iterations necessary to converge to a root can be shown through variety of color intensities. Points calling for fewer iterations appear with greater intensity. Through employing these tools, sensible geometric interpretations are possible for nearly all complex polynomials.

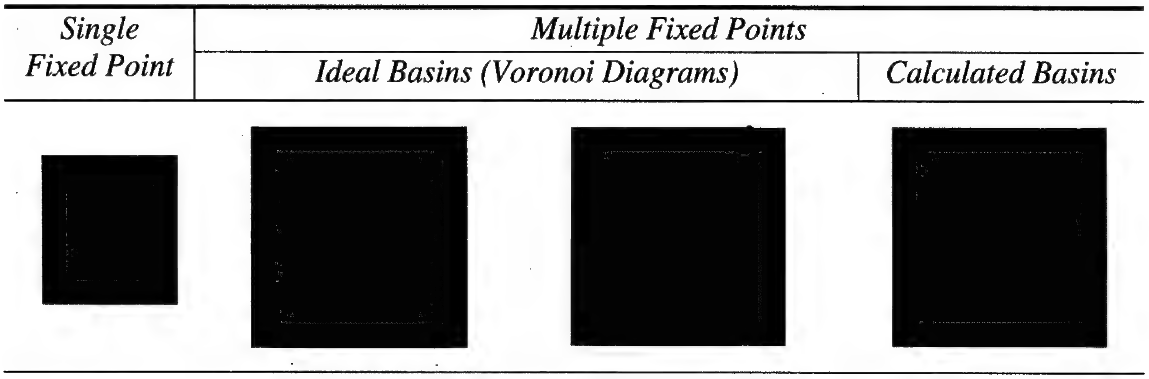


Figure 4.2. Basins of Attraction Coloring

A. PURE REAL AND PURE IMAGINARY ROOTS

When considering pure real or pure imaginary roots, the geometries, while still creating a variety of basin shapes, sizes, and complexities, are remarkably similar – only differing by a rotation to the appropriate coordinate the axes. Consequently, observations for one case support the other.

Even in what appears to be the simplest of settings, real roots, our basins of attraction are not ideal (Figure 4.3). Whether considering the case of equally or unequally distributed roots (Figures 4.4 – 4.9), nearest neighbor convergence fails to hold. With the exception of Laguerre's method, the shape of the basins roughly appears to conform to that of a hyperbola. Laguerre's basin, on the other hand, resembles our anticipated basin shape to a minor degree – especially for those eager to see some relationship. Symmetry is another common factor that plays a role in shaping the basins. Equally distributed roots generate symmetry throughout the geometry; unequally distributed roots do not.

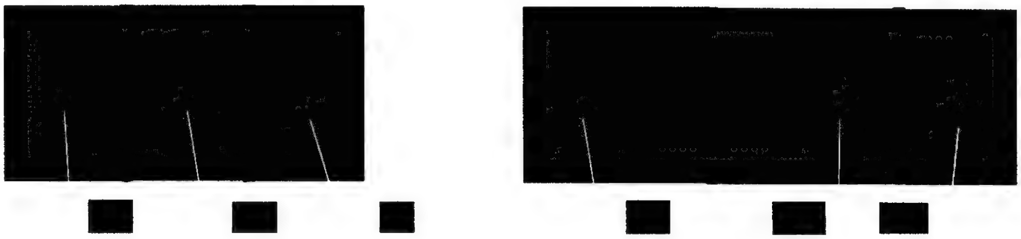
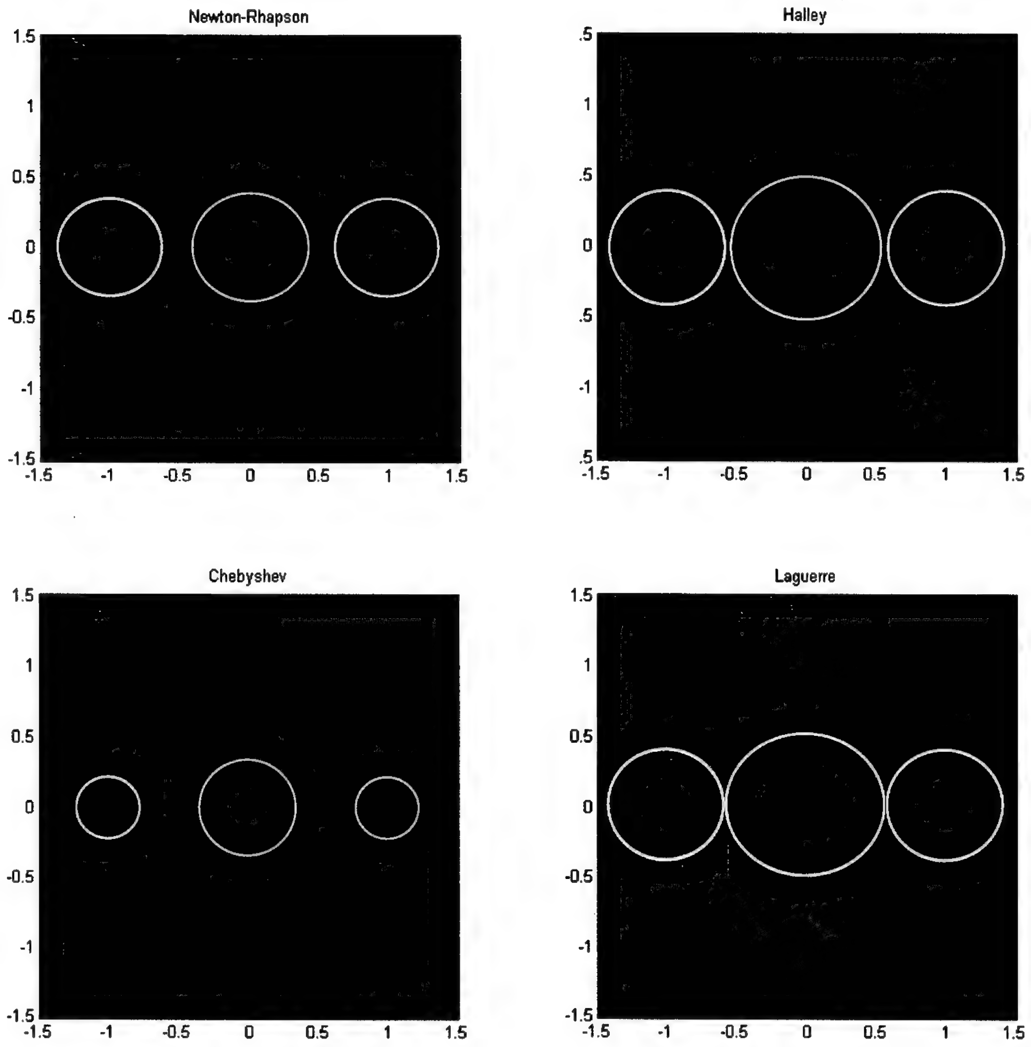


Figure 4.3. Ideal Basins & Associated Roots

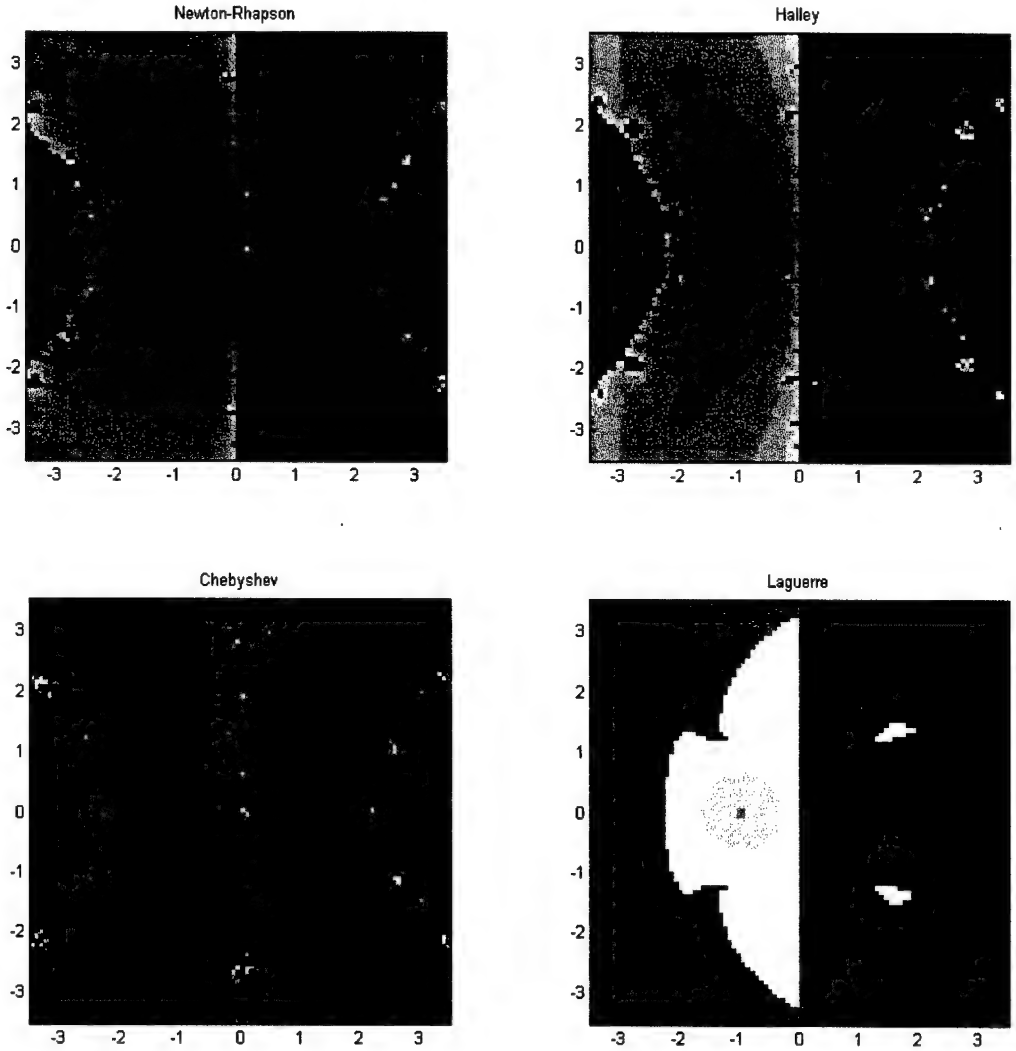
With slight exceptions along basin boundaries, the basin sizes for these are fairly comparable. For each, there exists some *effective* radius of convergence – that is, points in the neighborhood of a root tend toward that particular root (Figure 4.4). As Figures 4.4 – 4.9 suggest, that effective convergence radius not only changes amongst the numerical methods applied, but also with each polynomial considered. With higher order polynomials commonly creating more and more complex geometries, such radii are often greatly reduced.

Each method also bears some sensitive dependence on starting conditions. With nearby starting points assuming distinct solution paths, unpredictable behaviors can result. Although Figure 2.7 pointed out the concept initially, chaos' impact cannot be ignored – particularly with it present in every method's geometry. Figure 4.10 further reveals that basin boundaries may be self-similar, with infinite levels of detail, characteristic of fractal geometry.



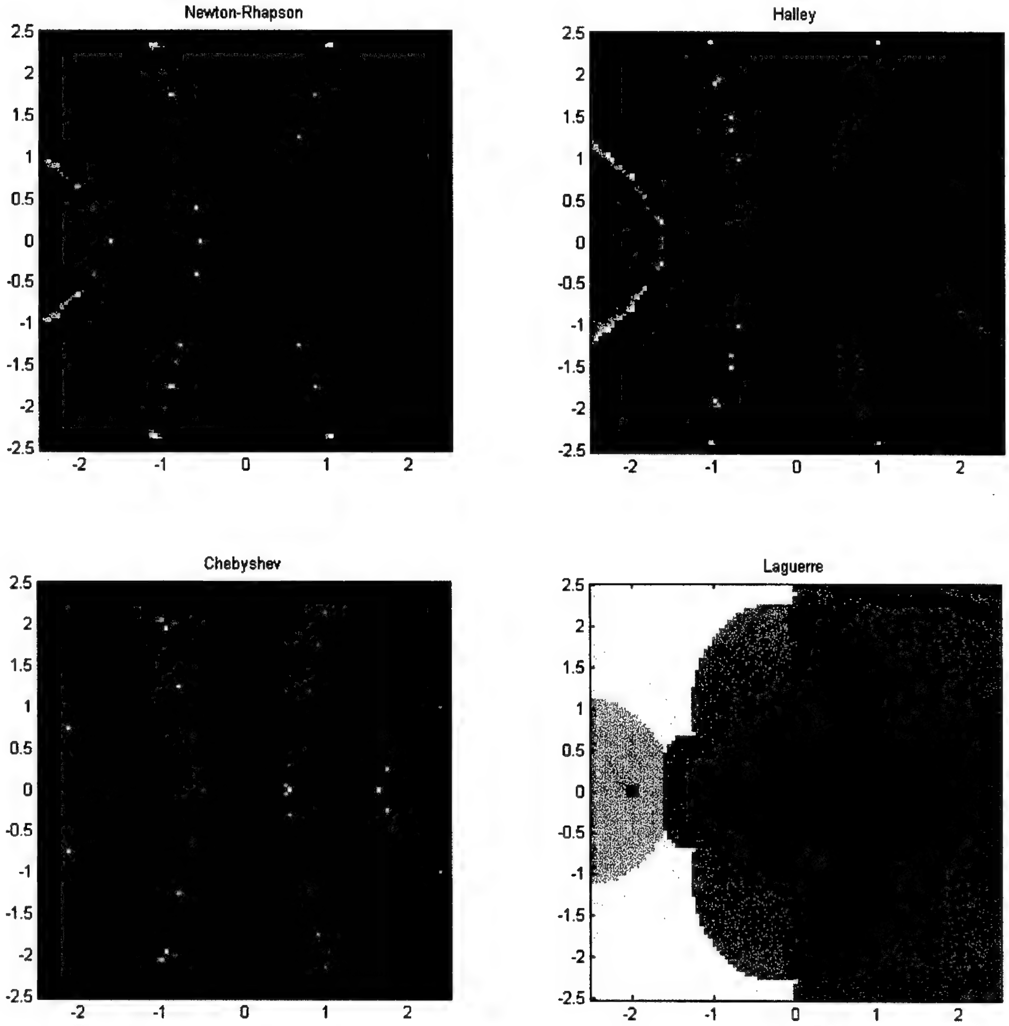
Roots
[-1, 0, 1]

Figure 4.4. Equally Spaced Roots – 3rd Order



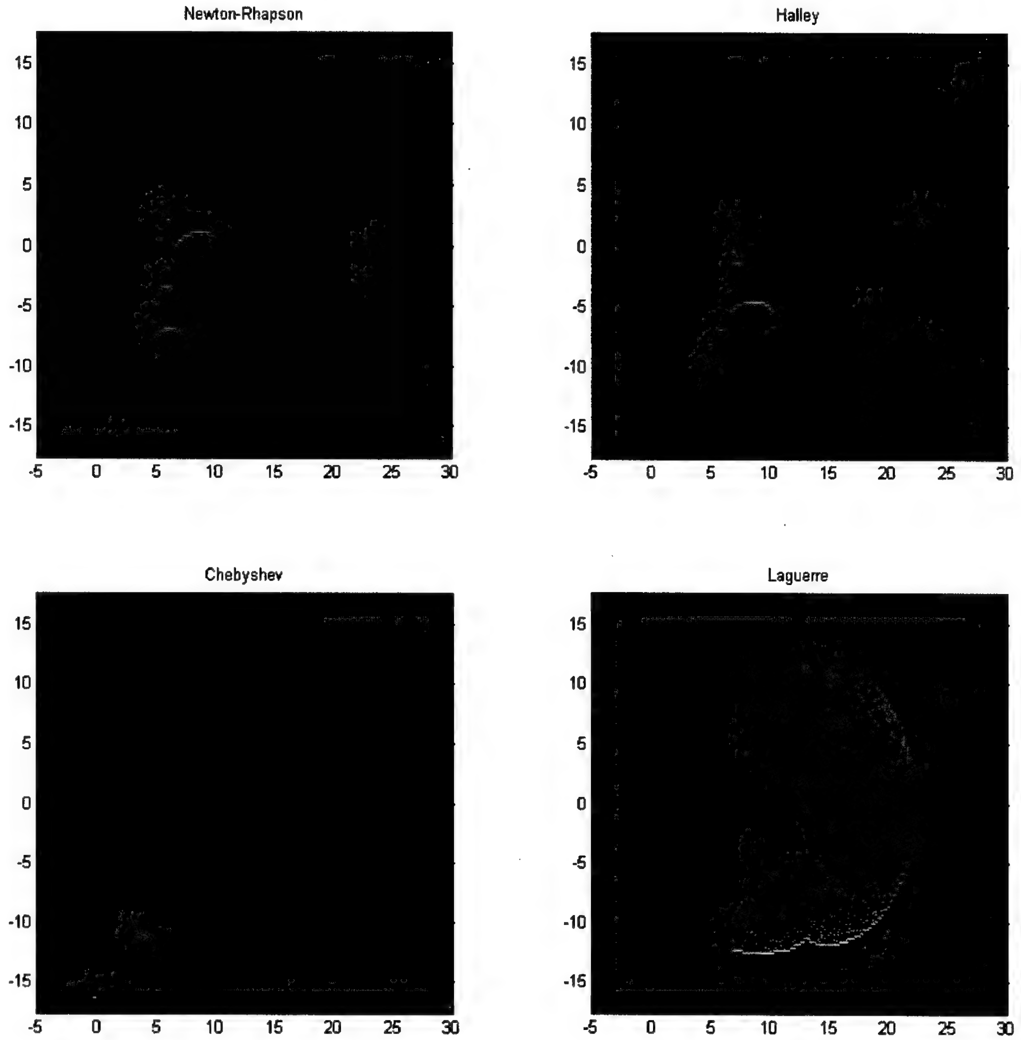
Roots
[-3, -1, 1, 3]

Figure 4.5. Equally Spaced Roots – 4th Order



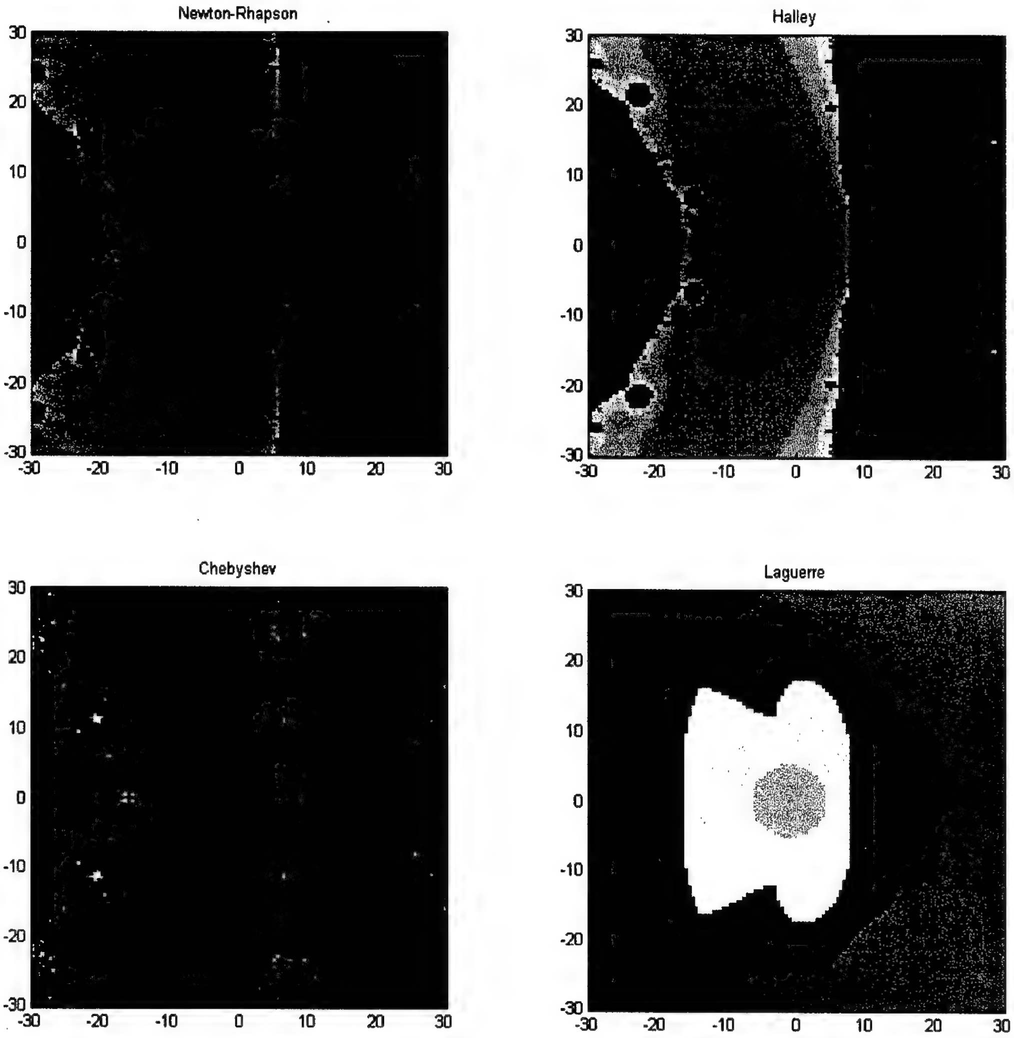
Roots
[-2 -1, 0, 1, 2]

Figure 4.6. Equally Spaced Roots – 5th Order



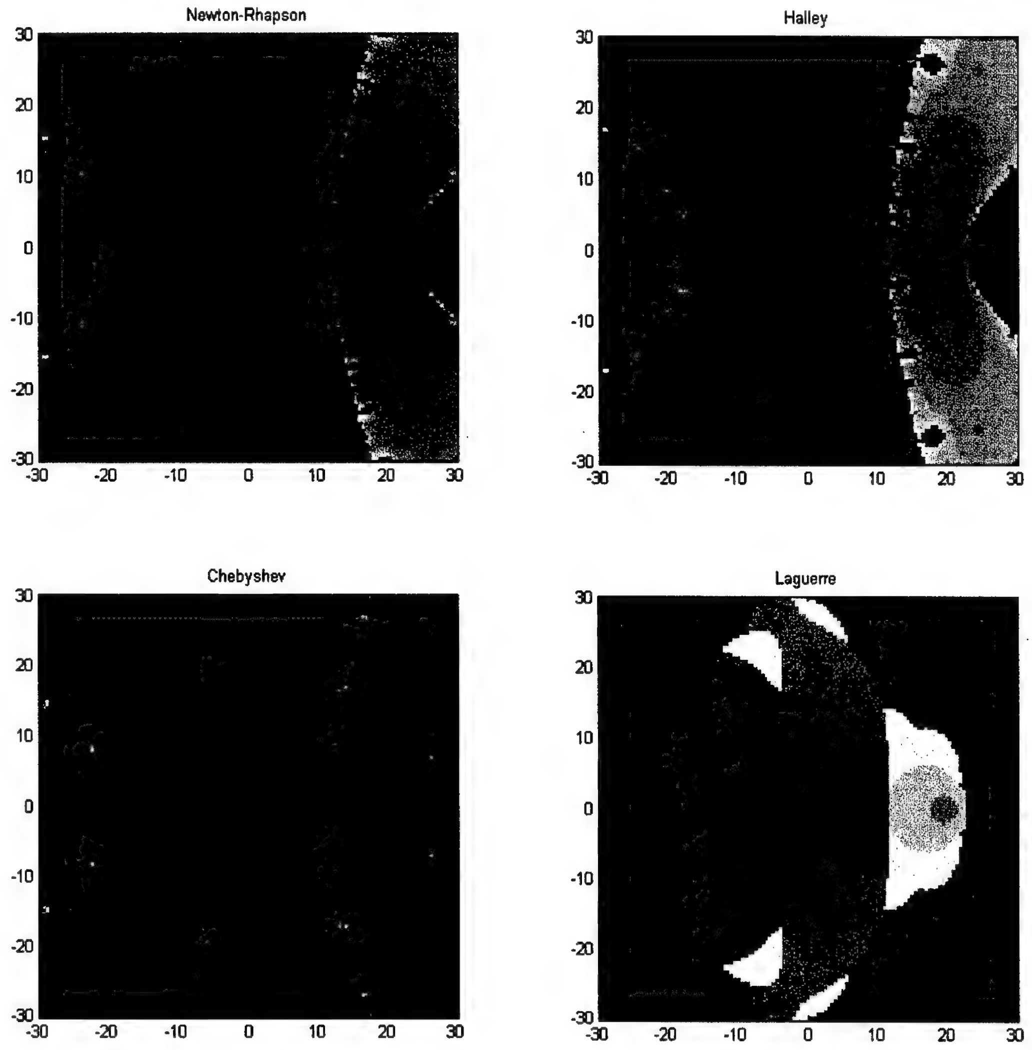
Roots
[-1, 20, 25]

Figure 4.7. Unequally Spaced Roots – 3rd Order



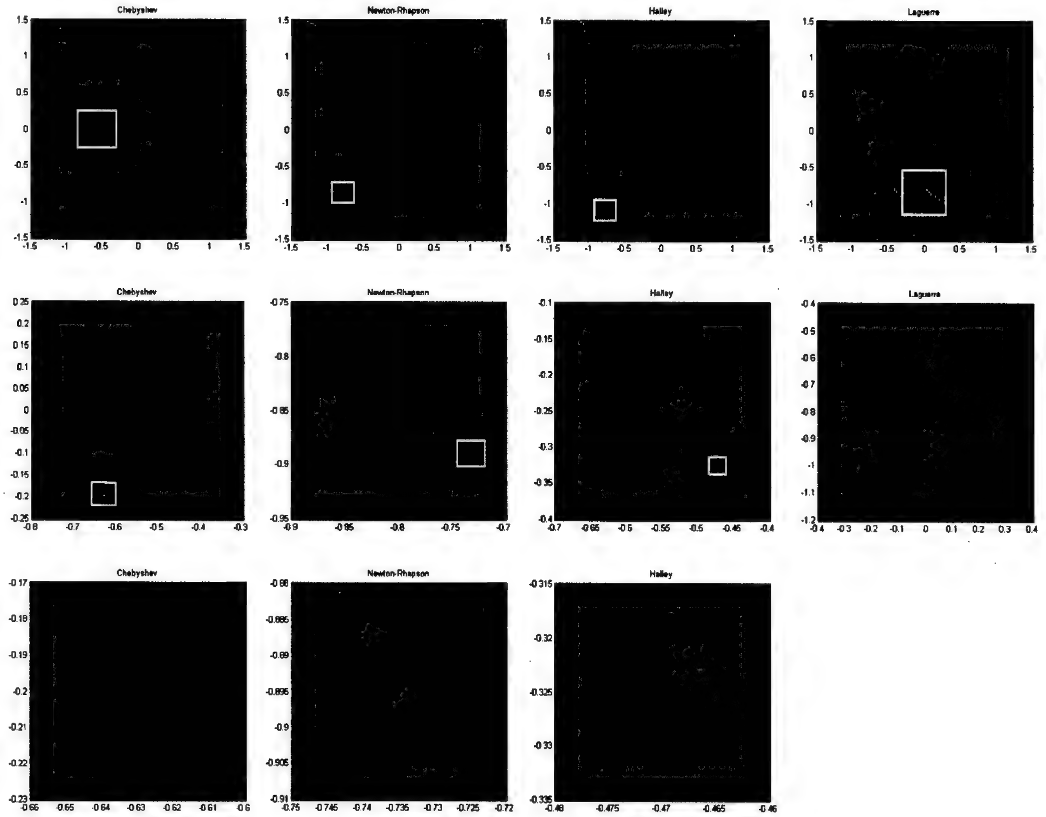
Roots
[-25, -1, 20, 25]

Figure 4.8. Unequally Spaced Roots – 4th Order



Roots
[-25 -1, 0, 20, 25]

Figure 4.9. Unequally Spaced Roots – 5th Order



<u>Roots</u>
[-1, 0, 1]

Figure 4.10. Chaos Everywhere

In considering the sensitive dependence on starting conditions, one need to only observe the '*decorations'* along the basin boundaries for each method's geometry in terms frequency, size, and structure. As a consequence of the competition and coexistence of more and more fixed points, the decorations appear with greater frequency with higher order polynomials, yet their size decreases. Whether equally or unequally spaced real roots, the Newton-Rhapson, Chebyshev and Halley methods appear to experience chaotic dynamics in a phase-shifted manner with each other – an unexpected outcome of their iterators. With rather clean, crisp boundaries, Laguerre's approximation technique provides better, though not absolute, predictability over the other methods.

While pure roots create nifty geometries, things get really interesting with mixed roots.

B. MIXED ROOTS

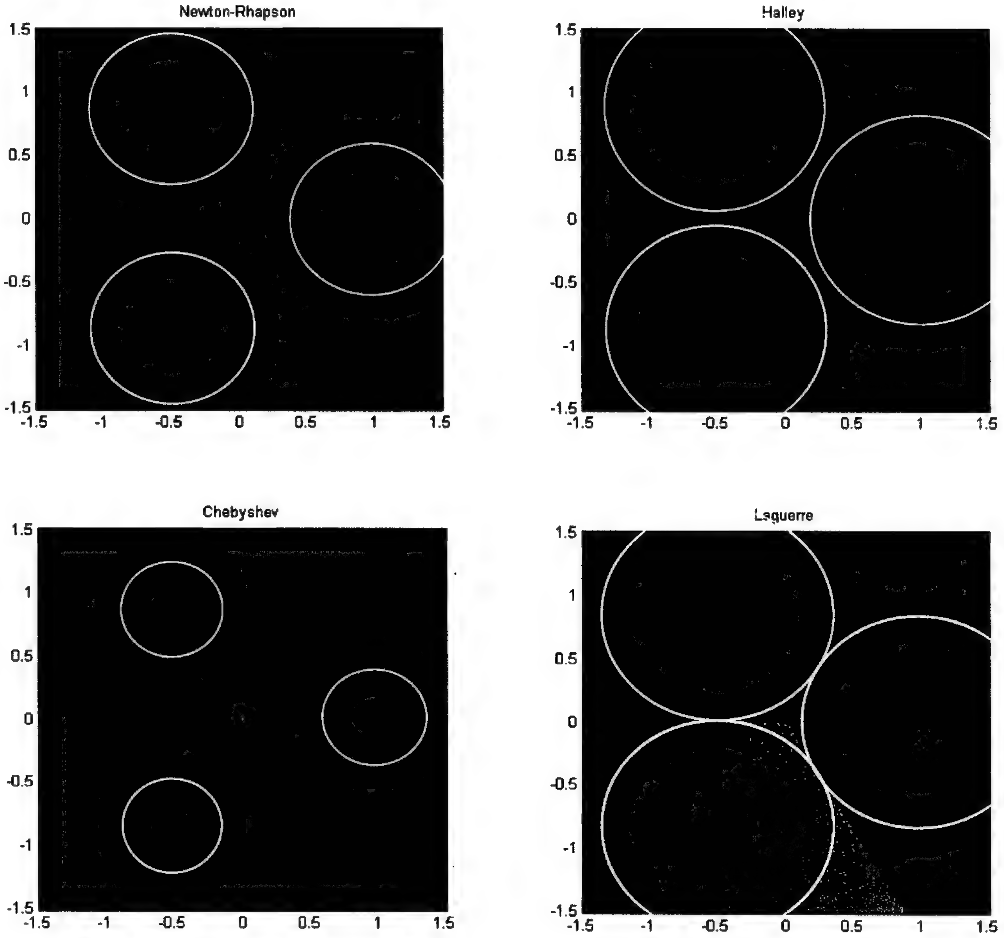
Mixed roots, those roots containing both real and imaginary components, provide a rich variety of basin shapes, sizes, and complexities. In many instances, there are striking similarities amongst these types of roots and pure ones. But when differences appear, a spectrum of spectacular geometries develops.

1. Roots of Unity (Equally Distributed Roots)

In the simplest of settings, roots of unity, there are more similarities than differences. Nearest neighbor convergence fails to hold, save Laguerre's approximation to the third order polynomial (Figure 4.11). As expected from the equal distribution of roots, basin shapes are symmetric (Figures 4.11 – 4.13). Again, these shapes lend to the equally distributed sizes of each basin.

Basin boundaries vary considerably – spanning from the very simple to the exceptionally intricate. Consequently, basins are more disconnected. As for sensitive dependence on starting conditions, the '*decorations*' for each method's geometry in terms frequency, size, and structure remains similar to the case of real roots. Furthermore, the effective radius of convergence is also affected accordingly (Figure 4.11). Figures 4.11 – 4.13 suggest, that effective convergence radius not only changes amongst the numerical methods applied, but also with each polynomial considered.

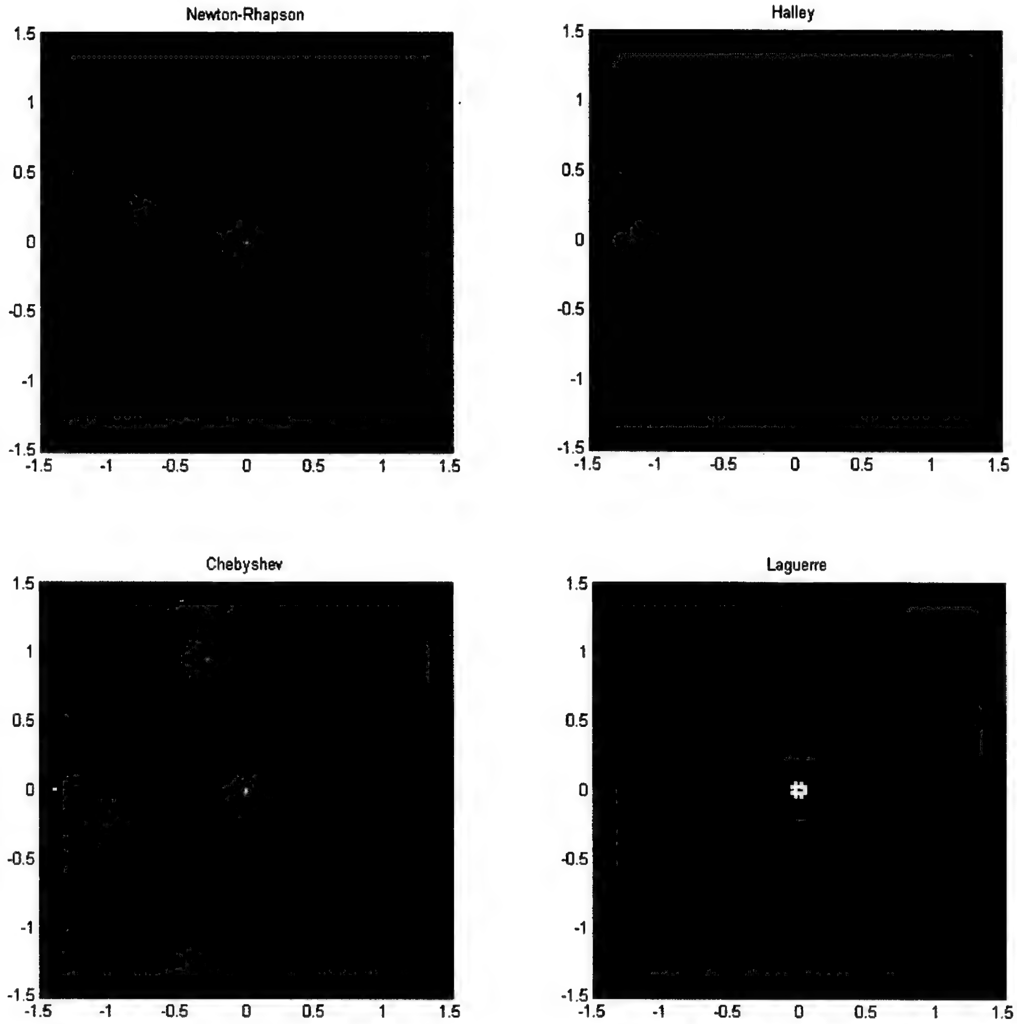
Again, as larger order polynomials are considered, basins become more complicated. Such behavior occurred previously, so this is of no great surprise. What is of great surprise is the rapid degradation of the Laguerre method near the origin (Figure 4.14). The apparent 'disks of chaos' seem analogous to Feigenbaum's universality – that qualitative changes leading from order to chaos and chaos into order exist (Peitgen, 1992).



<u>Roots</u>

$$\left[1,\ \frac{1}{2}+\frac{\sqrt{3}}{2}i,\ \frac{1}{2}-\frac{\sqrt{3}}{2}i\right]$$

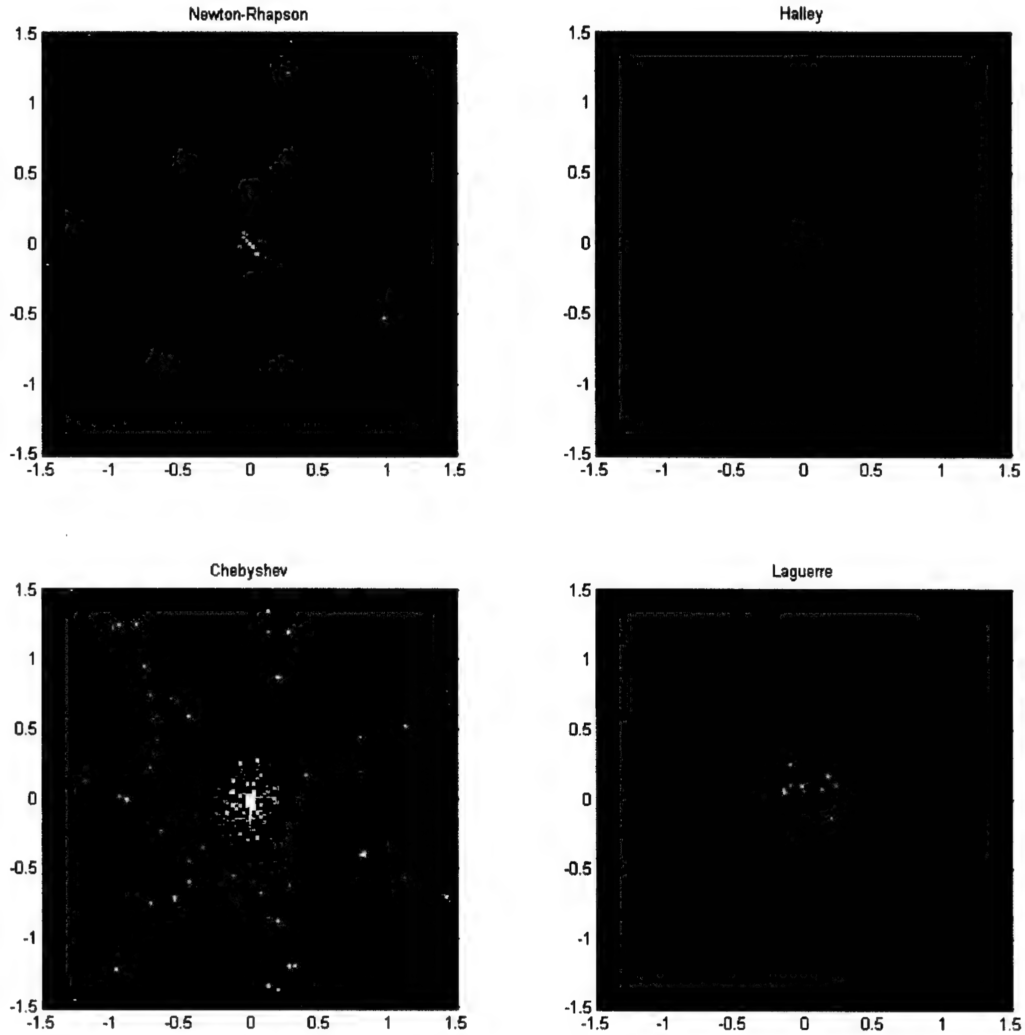
Figure 4.11. Roots of Unity – 3rd Order



Roots

$$\left[1,\ \frac{2}{5}\pm\frac{\pi}{10}i,\ \frac{4}{5}\pm\frac{3\pi}{10}i\right]$$

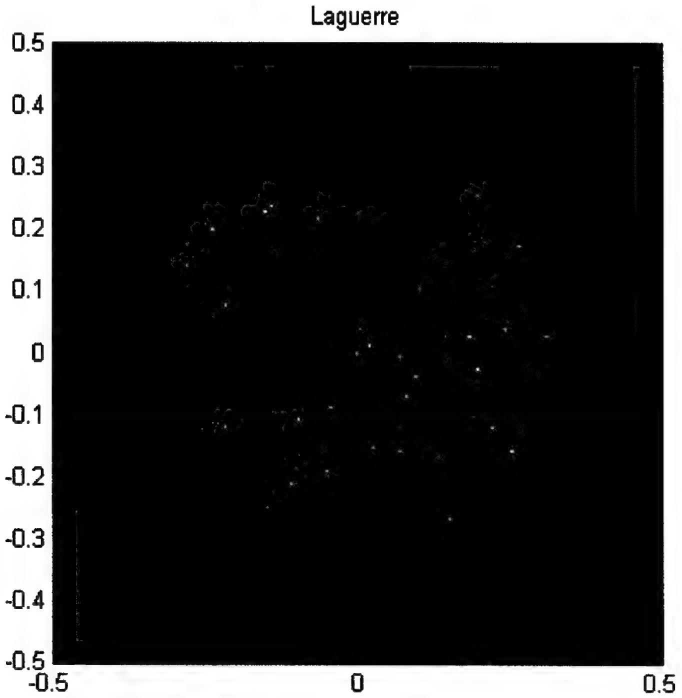
Figure 4.12. Roots of Unity – 5th Order



Roots

$$\left[1,\ ,-cos\left(\frac{\pi}{7}\right)\pm i\,sin\left(\frac{\pi}{7}\right),\ cos\left(\frac{2\pi}{7}\right)\pm i\,sin\left(\frac{2\pi}{7}\right),\ -cos\left(\frac{3\pi}{7}\right)\pm i\,sin\left(\frac{3\pi}{7}\right)\right]$$

Figure 4.13. Roots of Unity – 7th Order



Roots

$$\left[1,\ -\cos\left(\frac{\pi}{7}\right) \pm i\sin\left(\frac{\pi}{7}\right),\ \cos\left(\frac{2\pi}{7}\right) \pm i\sin\left(\frac{2\pi}{7}\right),\ -\cos\left(\frac{3\pi}{7}\right) \pm i\sin\left(\frac{3\pi}{7}\right)\right]$$

Figure 4.14. Roots of Unity – 7th Order (Zoom)

2. Unequally Distributed Roots

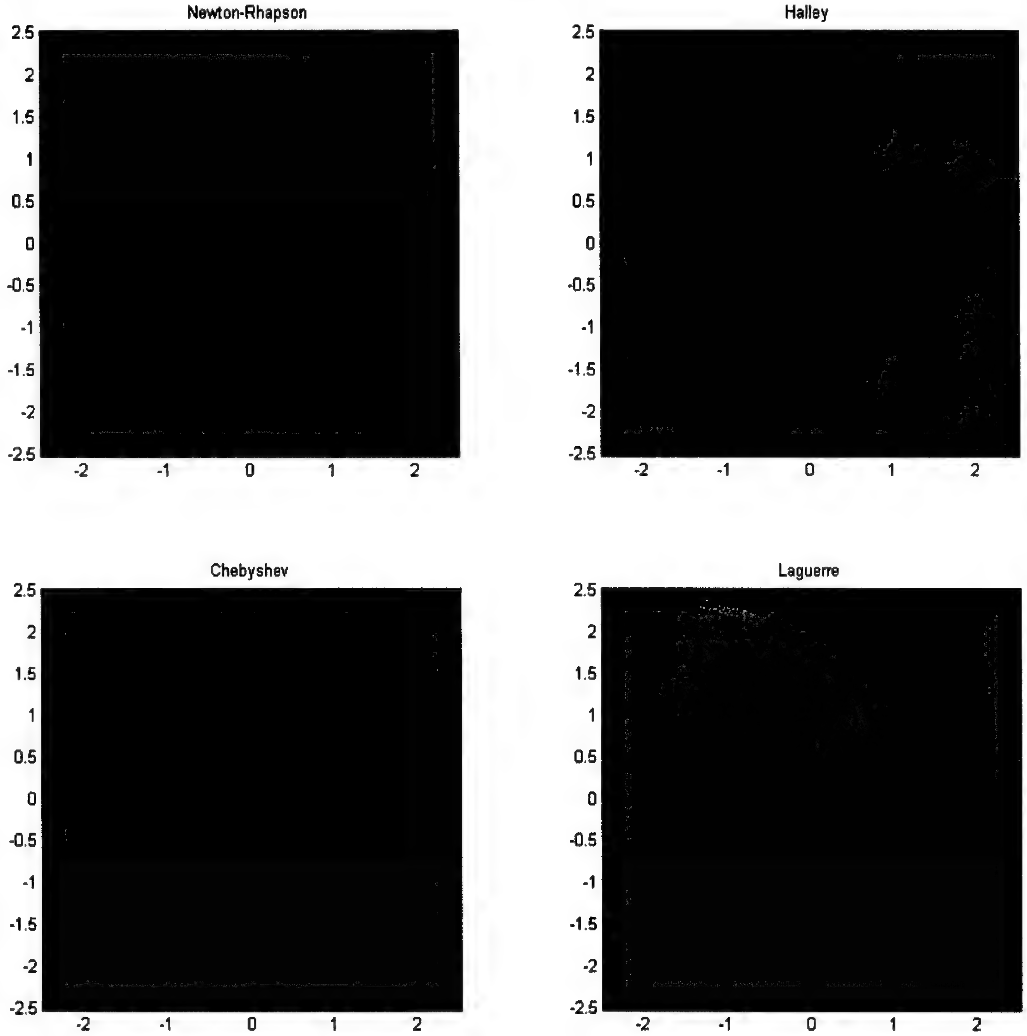
When roots are positioned irregularly, interesting things can and do happen. In general, the concepts of nearest neighbor convergence failing, basin shape, size, and complexity are as noted previously – but perhaps in a more pronounced fashion.

Third order polynomials produce a variety of familiar behaviors. While Figures 4.15 & 4.16 echo the behaviors previously found with real roots, Figures 4.17 & 4.18 behave similarly to the roots of unity. Why the difference? Simply put, one is nothing more than a skewed version of the other. When one fixed point is a *'near-enough'* reflection of another, a *'near-enough'* symmetric geometry results; otherwise, a distorted geometry develops. Preference for a particular numerical method is subject to the presence of such symmetry.

Figures 4.19 – 4.24 reveal how convergence changes with the next order of polynomials. With Figure 4.19 roots lying on a straight line, it is nothing more than a rotation of Figure 4.5, and it assumes similar behaviors. Figures 4.20 – 4.22, while containing a reflection of a fixed point to another, contain irregular and unexpected basins shapes and decorations – resulting from the competition and coexistence of a fourth fixed point. Halley's method generates the most pronounced irregularities, to include distinct breaks in basin connectivity. Figures 4.23 & 4.24 yield expected behaviors, save Halley.

As for higher order polynomials, Figures 4.25 – 4.33 depict various geometries for fifth and sixth order polynomials. While the geometries are qualitatively different, their interpretation is found through the application of previous geometry's behaviors (Figures 4.4 – 4.24).

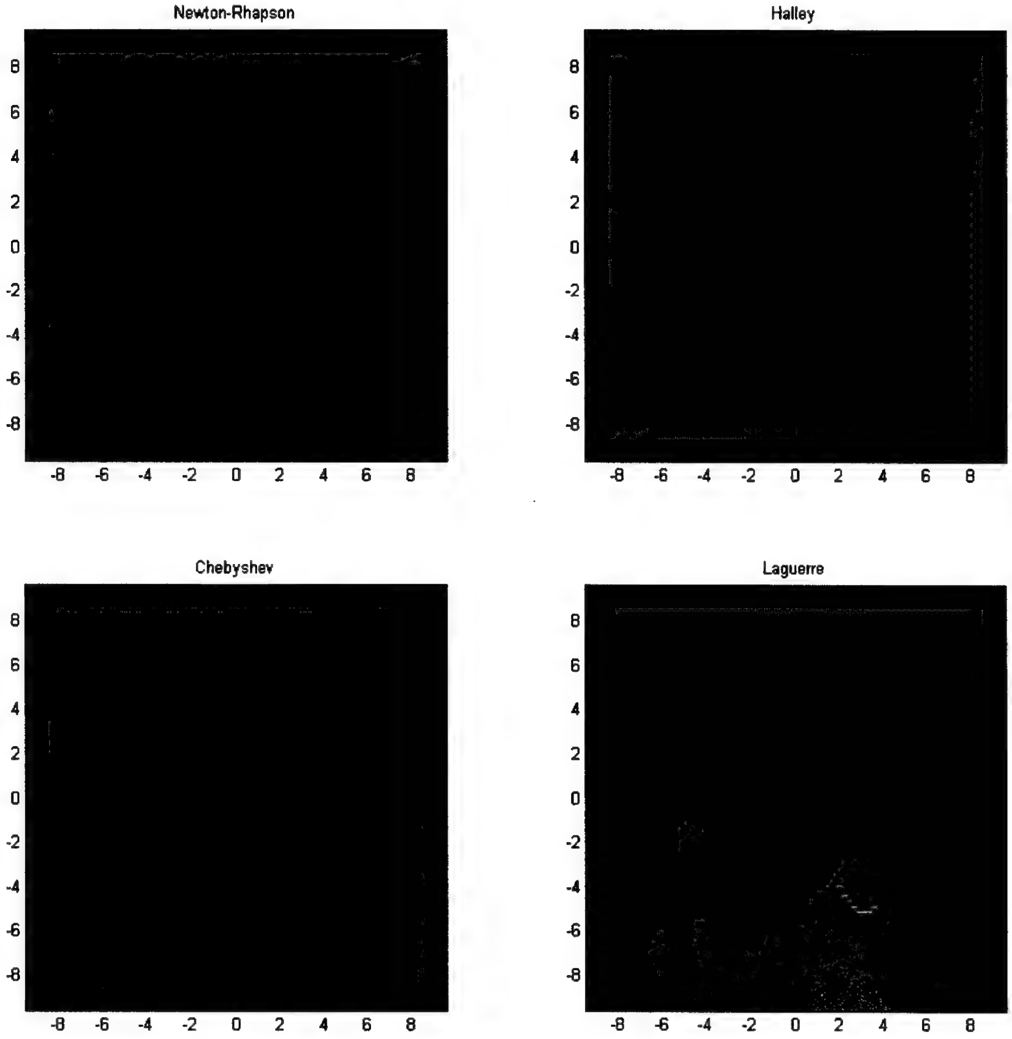
In all these instances, basin shapes, sizes and complexities vary considerably. And in nearly every case, the geometry remains unpredictable. But within this chaotic environment, is there any order?



Roots

$$\left[-1,\ -\frac{1}{2}+i,\ -\frac{1}{2}+2i\right]$$

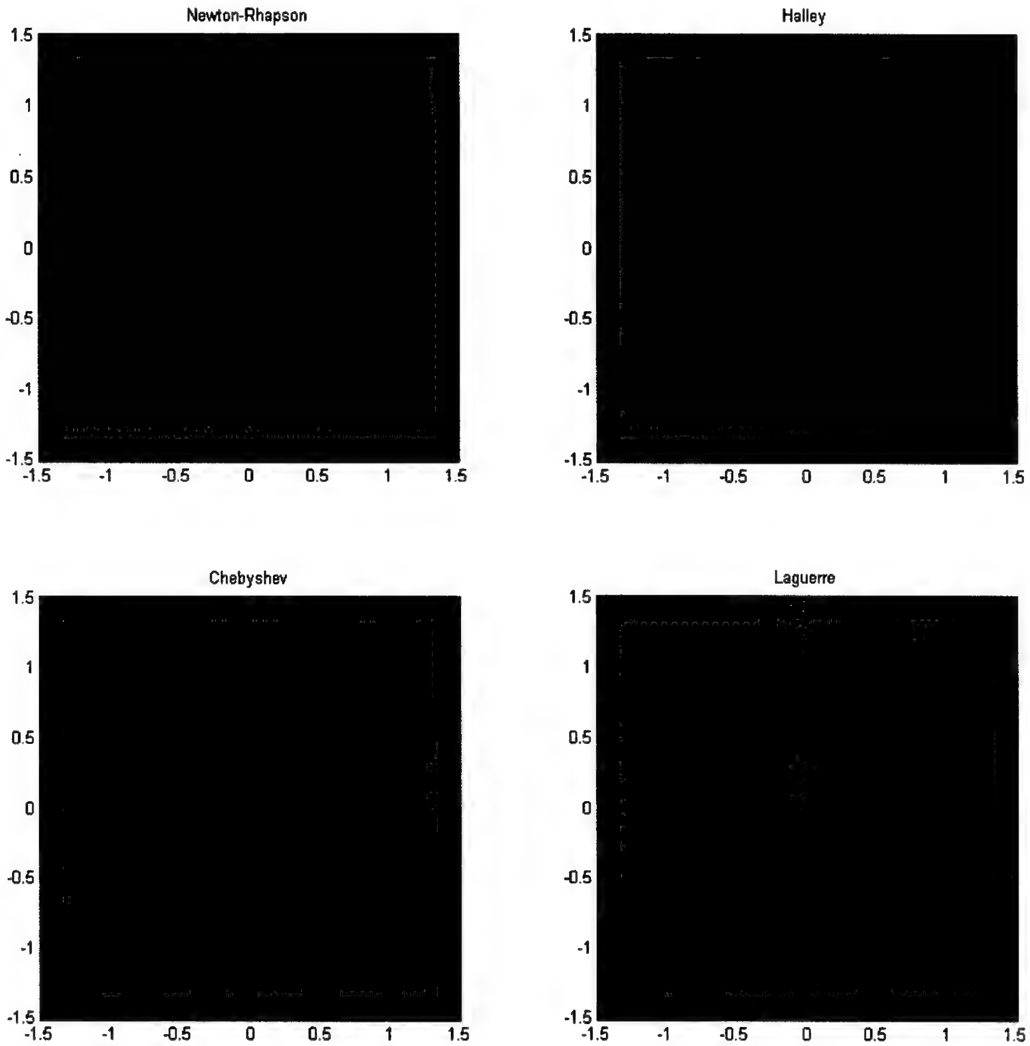
Figure 4.15. Mixed Roots – 3rd Order



<u>Roots</u>

$$\left[9,\ i,\ -6-6i\right]$$

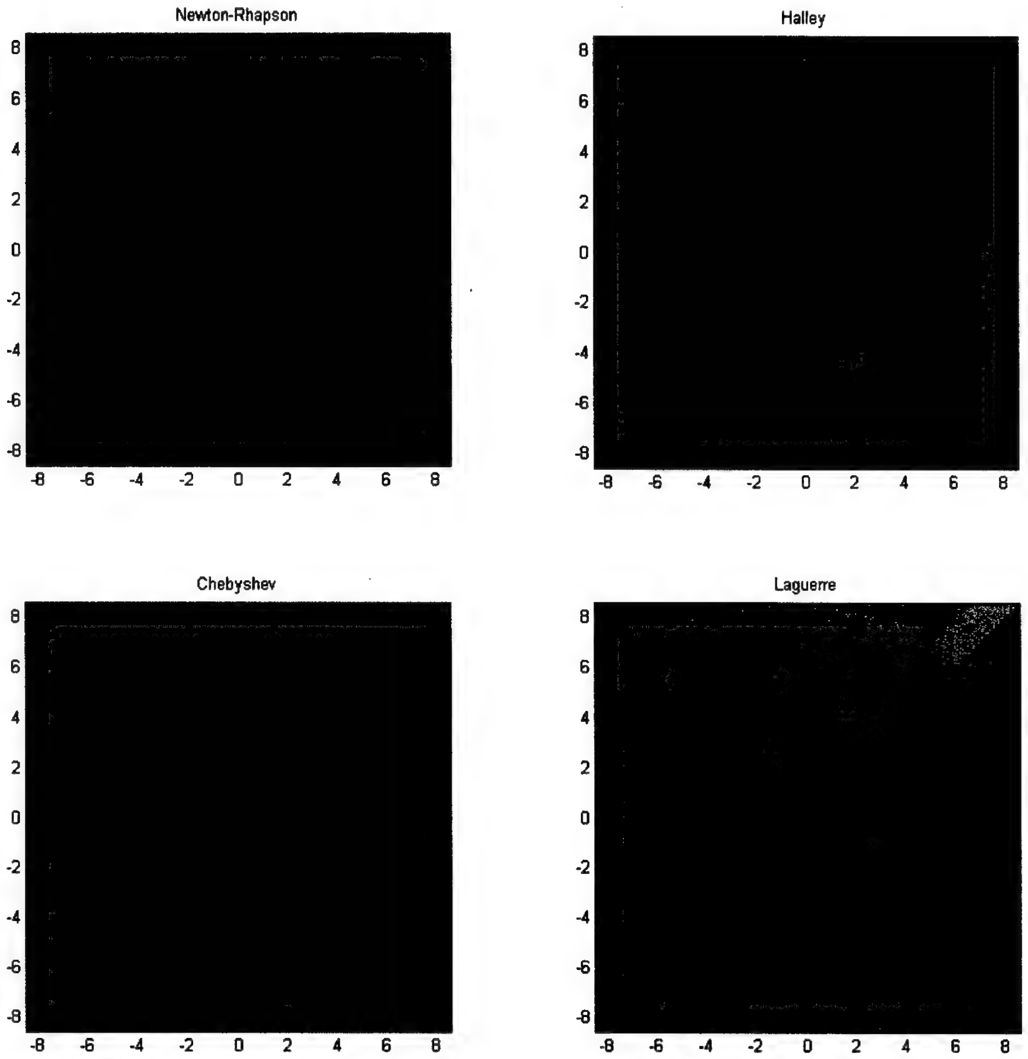
Figure 4.16. Mixed Roots – 3rd Order



Roots

$$\left[-i,\ ,\pm 1+i\right]$$

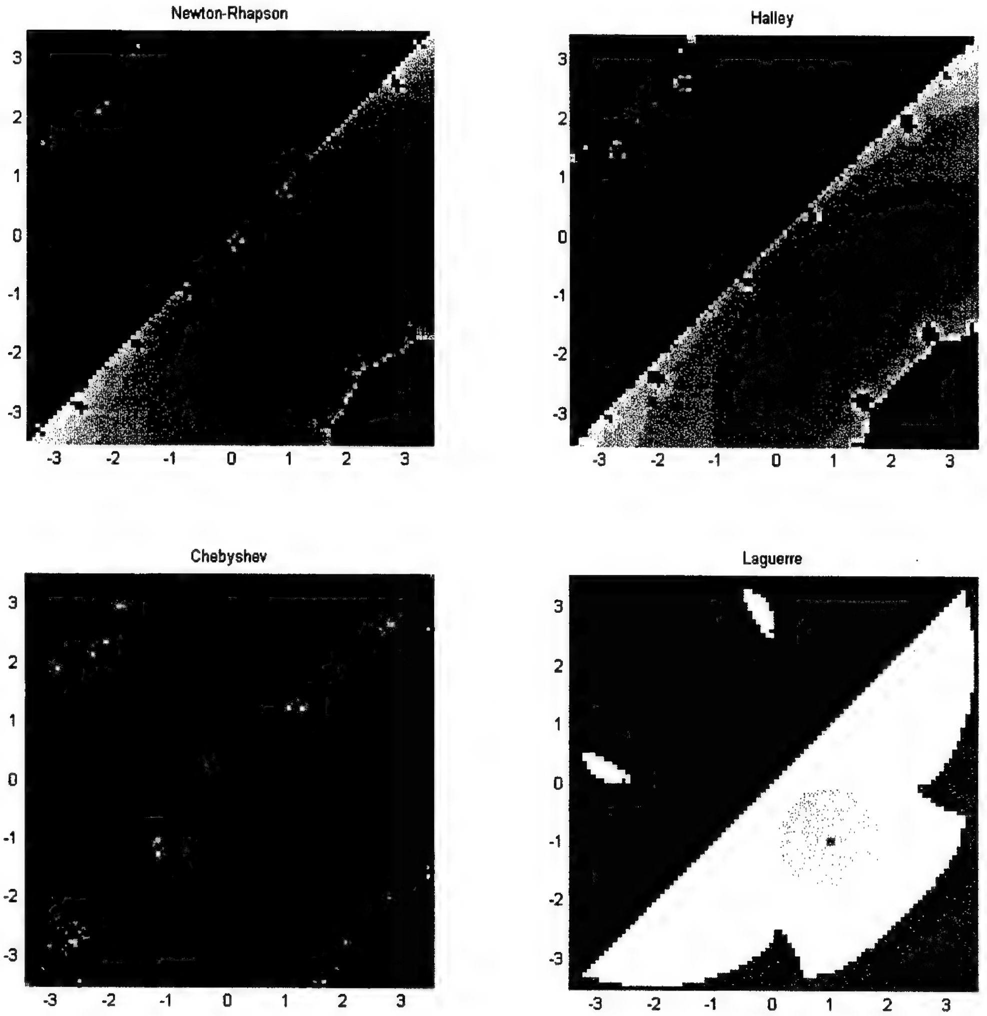
Figure 4.17. Mixed Roots – 3rd Order



Roots

$$\left[8,\ 8i,\ -4-4i\right]$$

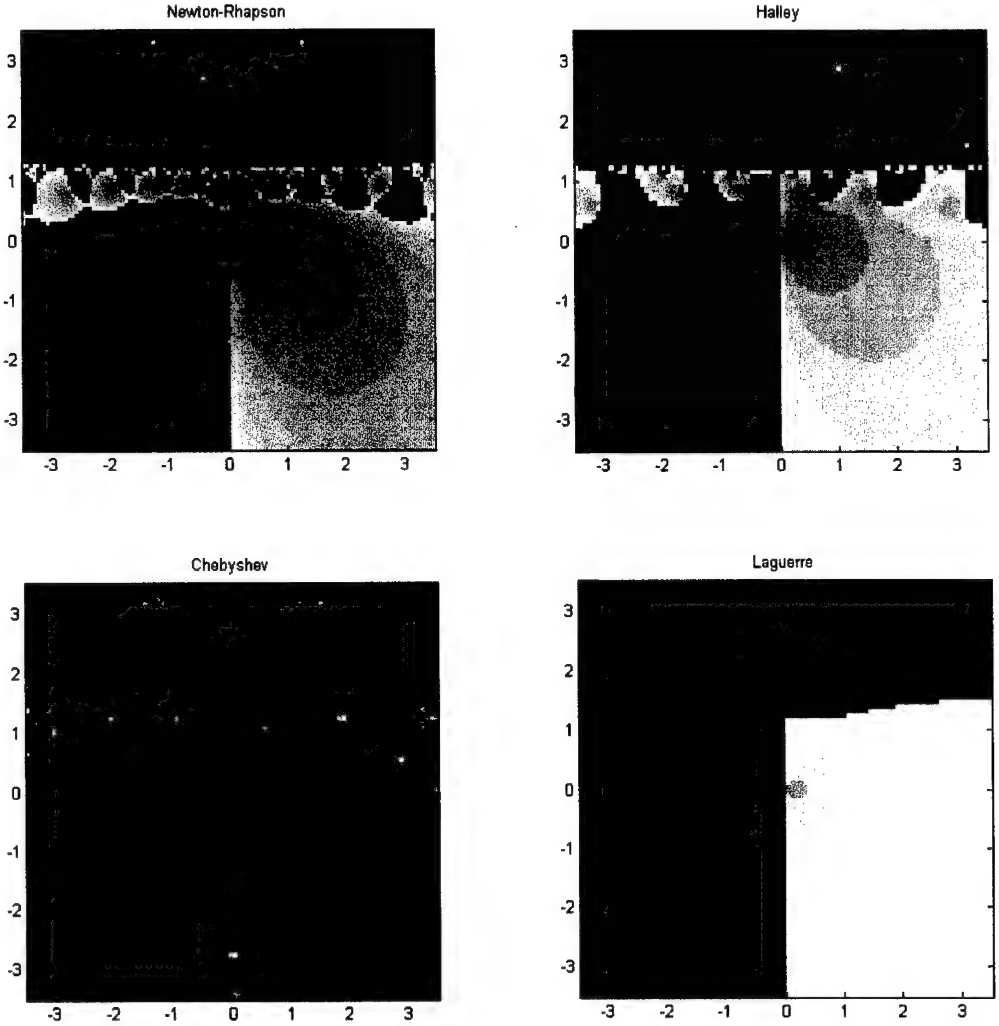
Figure 4.18. Mixed Roots – 3rd Order



Roots

$$\left[-3+3i,\ -1+i,\ 1-i,\ 3-3i\right]$$

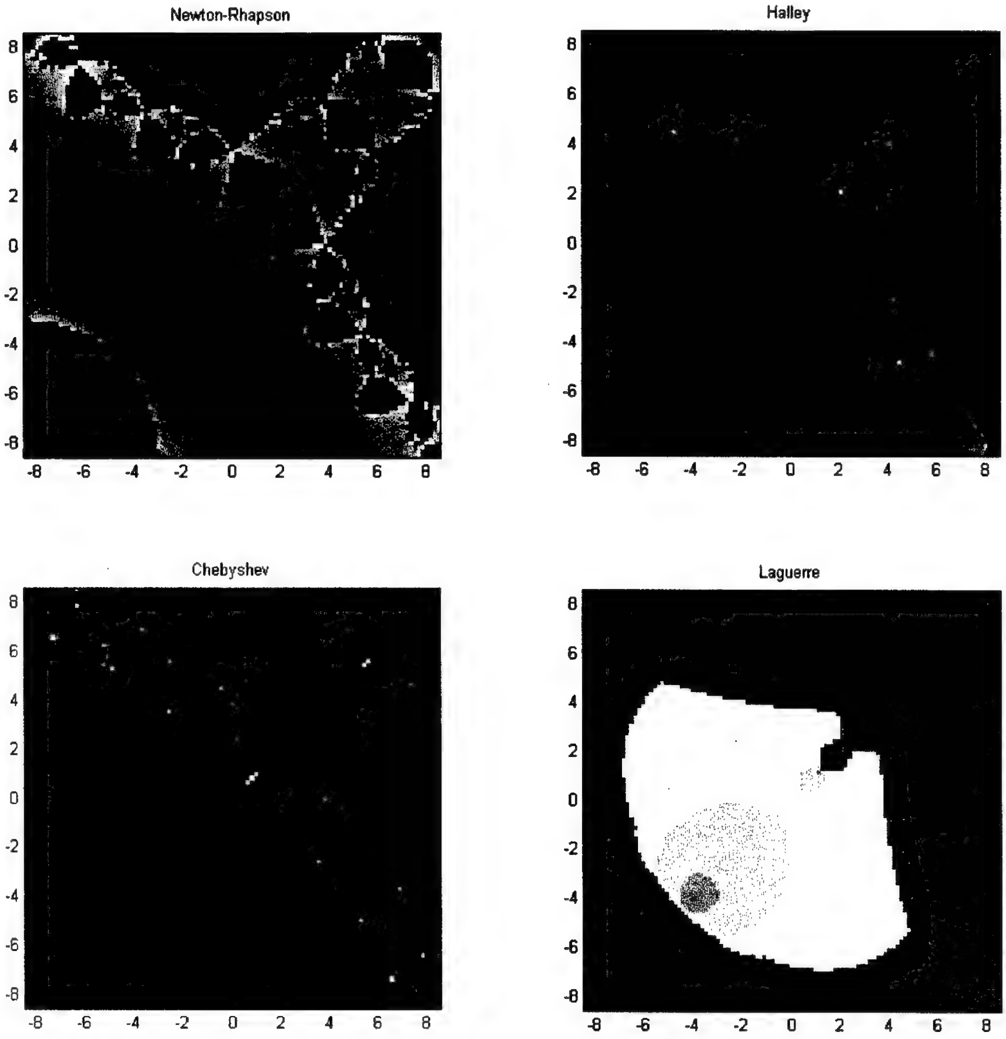
Figure 4.19. Mixed Roots – 4th Order



Roots

$$\left[\frac{1}{10}, \; , -\frac{1}{10}, \; 2i, \; 3i\right]$$

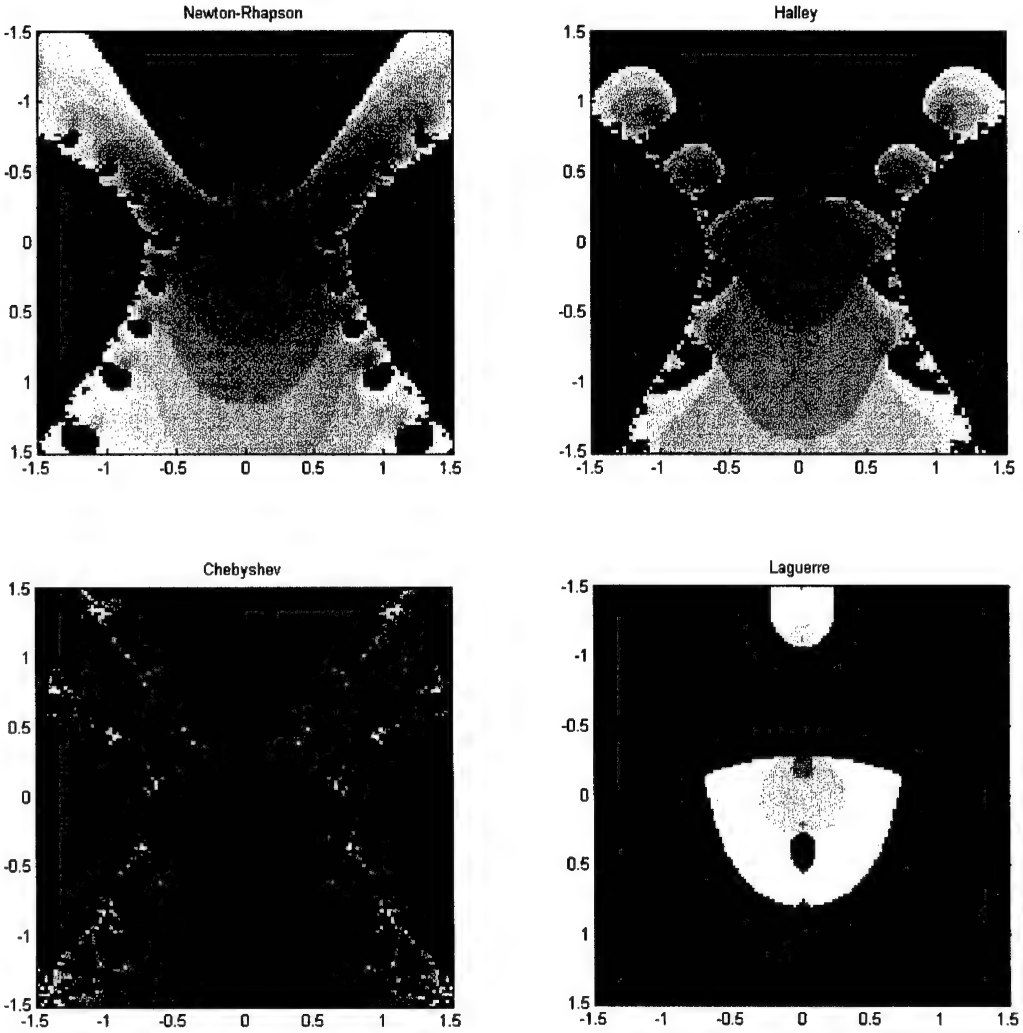
Figure 4.20. Mixed Roots – 4th Order



Roots

$$\left[8,\ 8i,\ -4-4i,\ -5-5i\right]$$

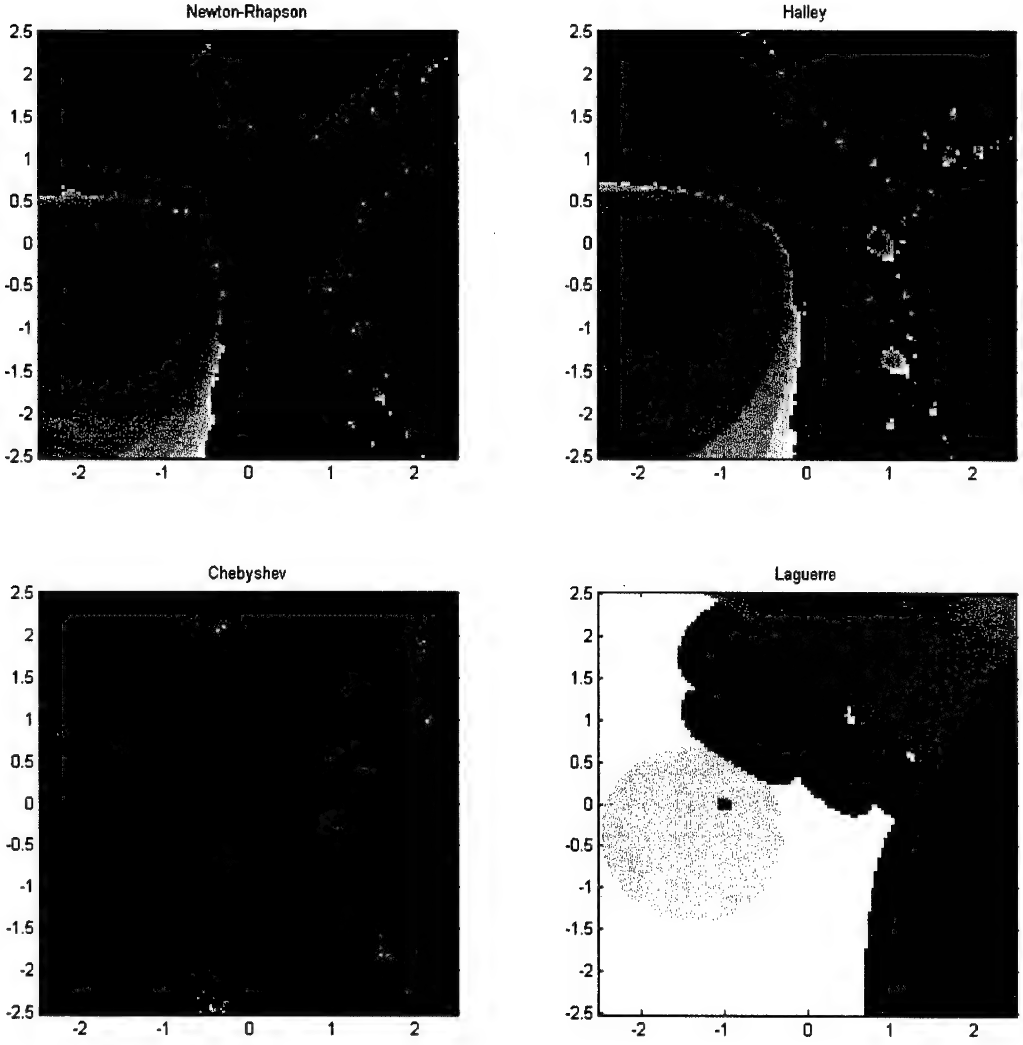
Figure 4.21. Mixed Roots – 4th Order



Roots

$$\left[\pm 1, \ ,\frac{1}{4}i \ ,\frac{1}{3}i\right]$$

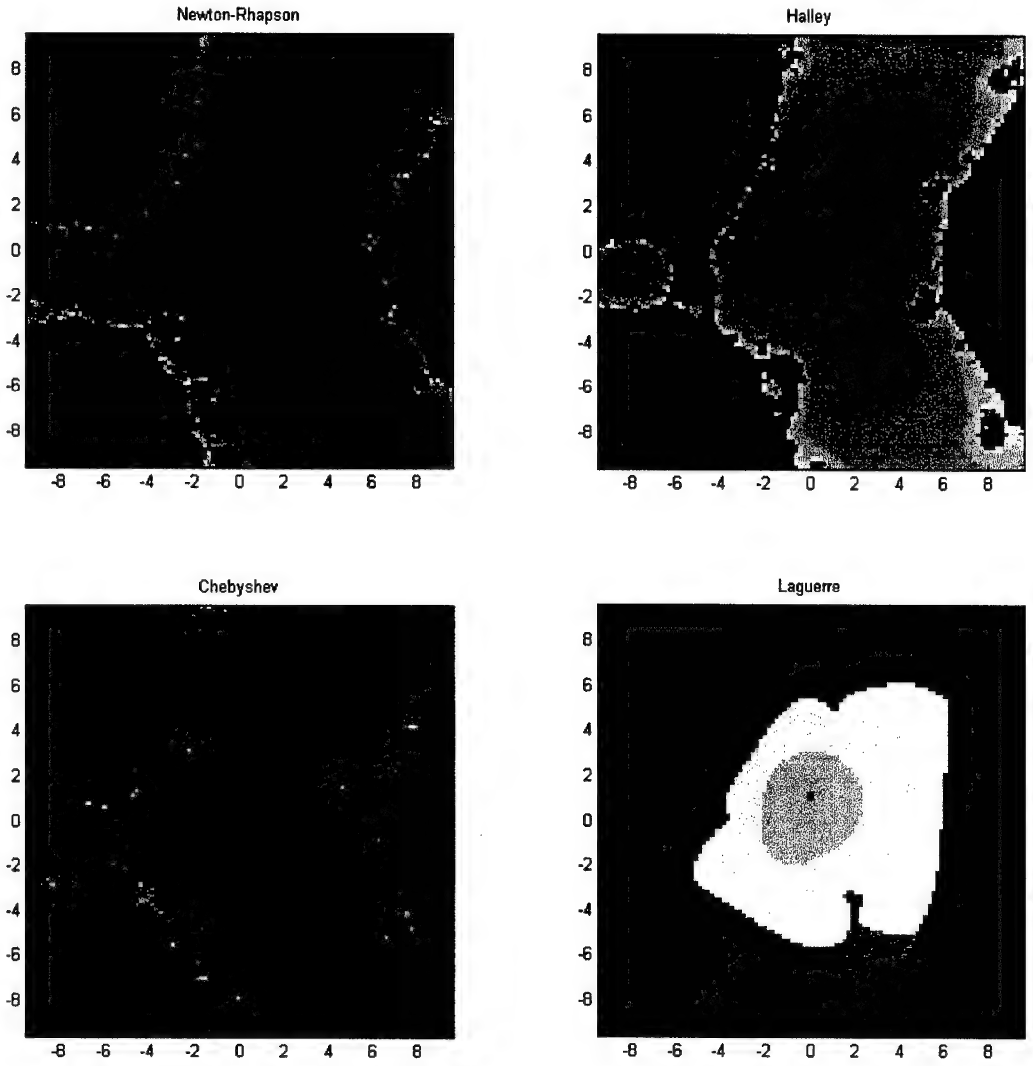
Figure 4.22. Mixed Roots – 4th Order



Roots

$$\left[-1,\ -\frac{1}{2}+i,\ -\frac{1}{2}+2i,\ 2\right]$$

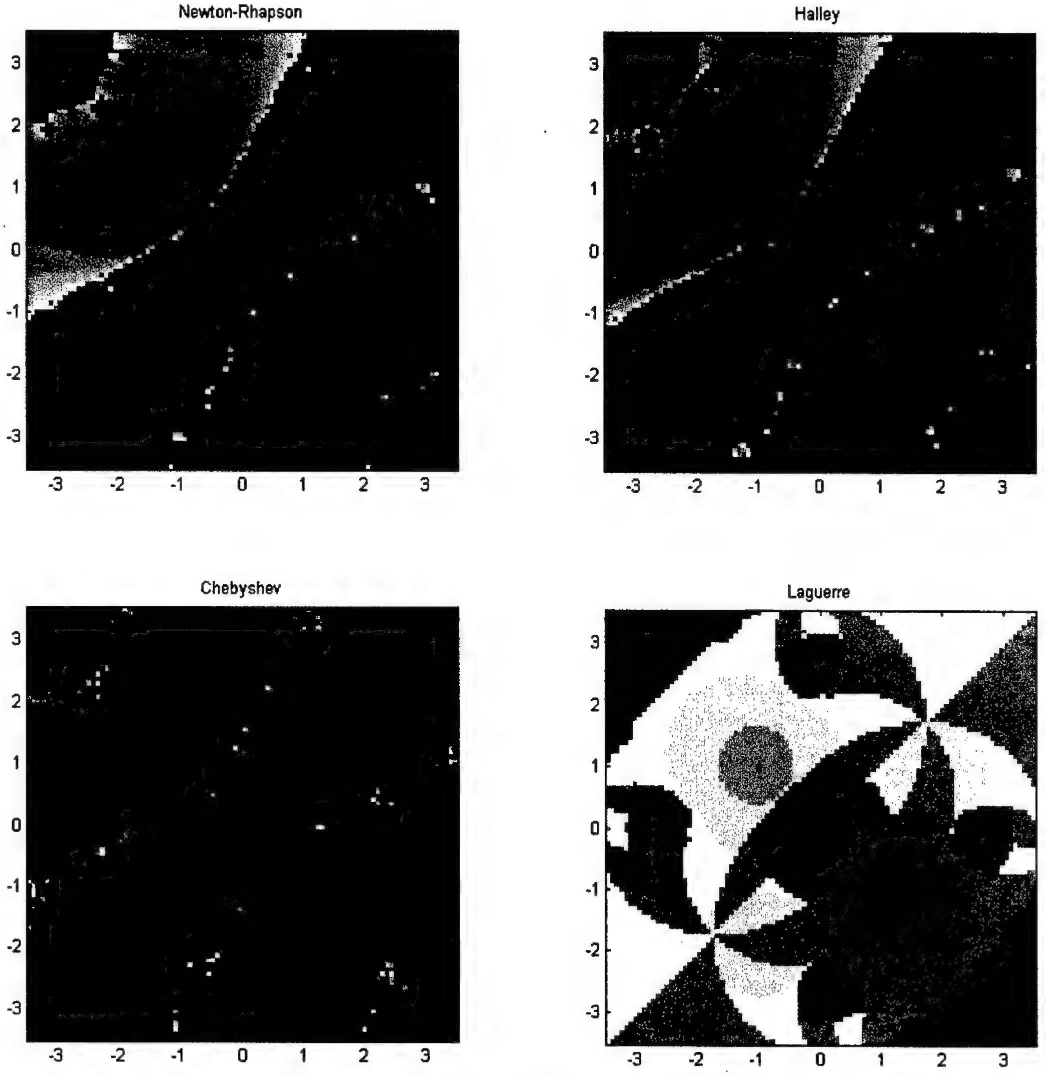
Figure 4.23. Mixed Roots – 4th Order



<u>Roots</u>

$$\left[9,\ i,\ -6-6i,\ -5+4i\right]$$

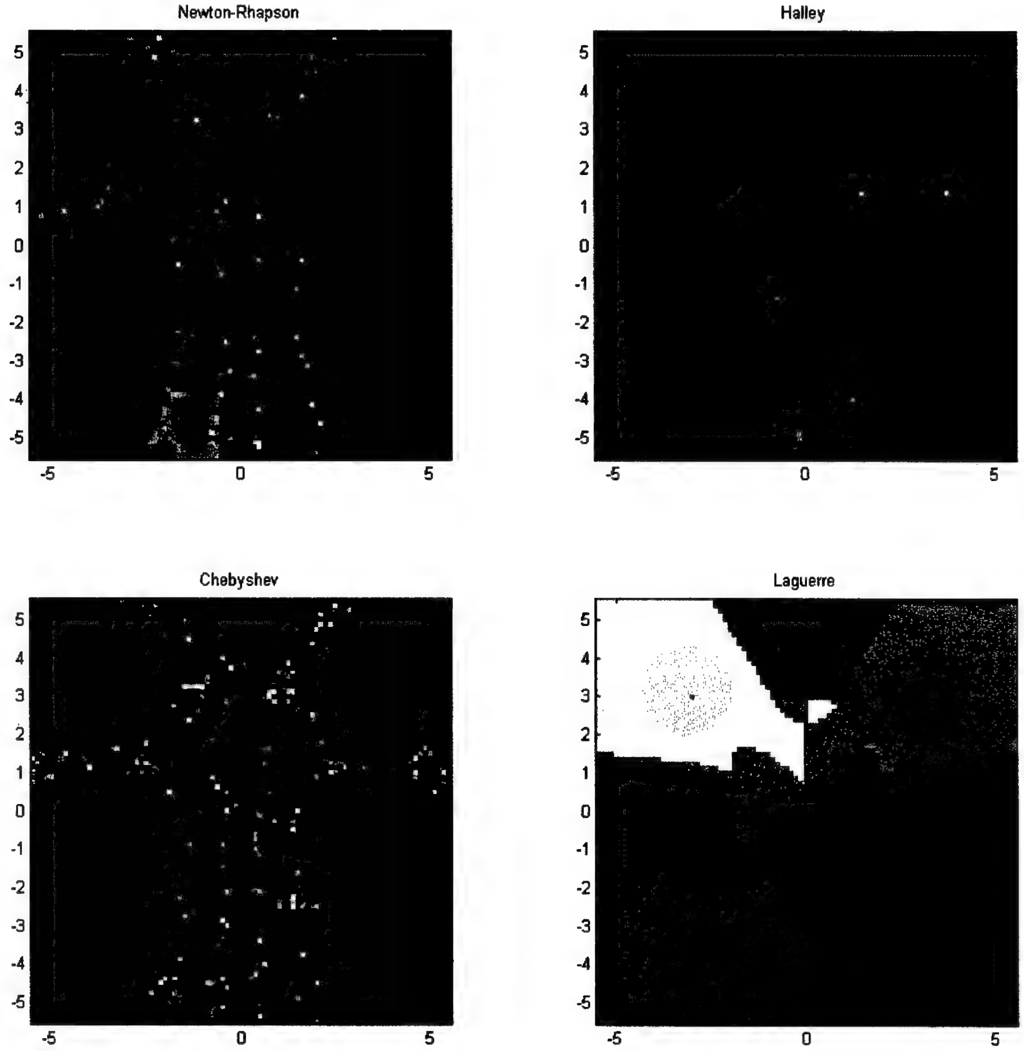
Figure 4.24. Mixed Roots – 4th Order



Roots

$$\left[-3+3i,\ -1+i,\ 0,\ 1-i,\ 3-3i \right]$$

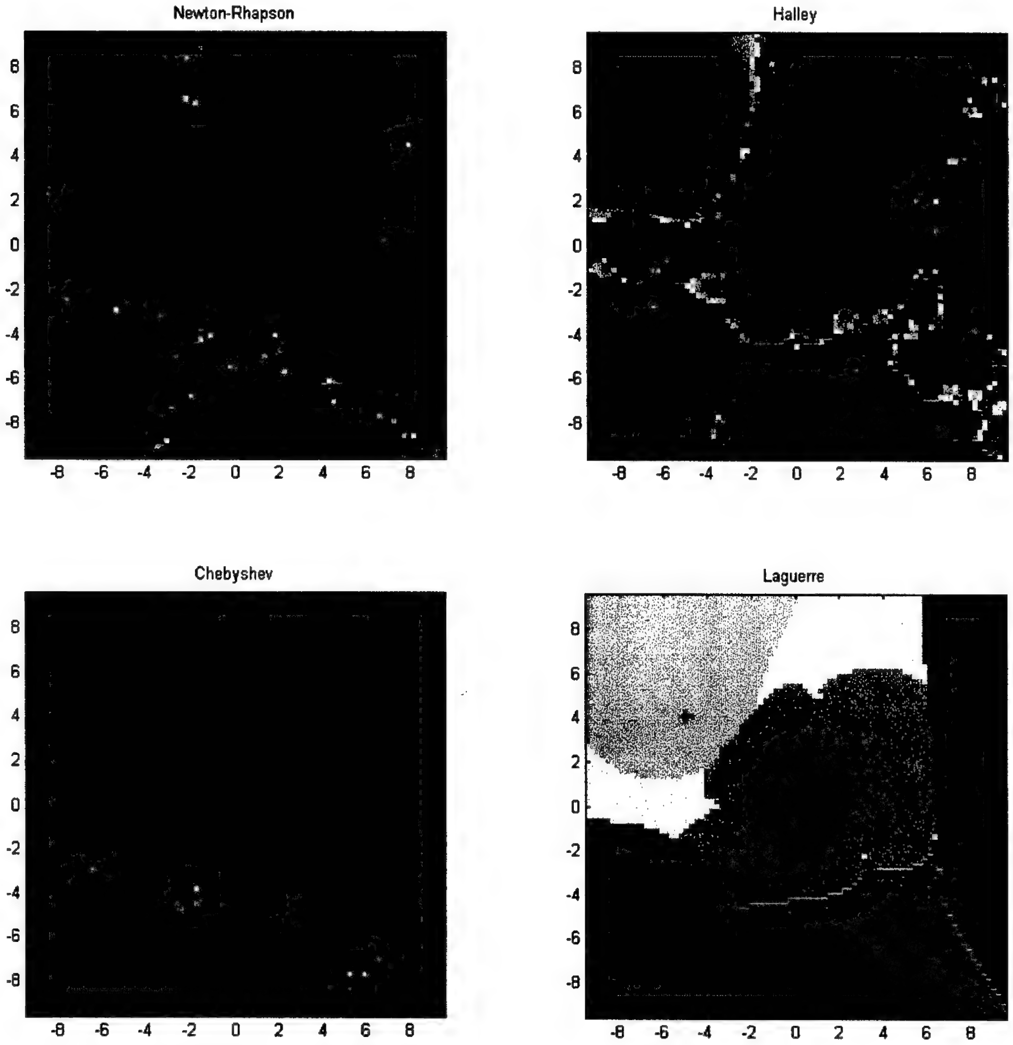
Figure 4.25. 5th Order Mixed Roots



<u>Roots</u>

$$\left[\pm 3,\ \ 5i\ \ ,\pm 3+3i\right]$$

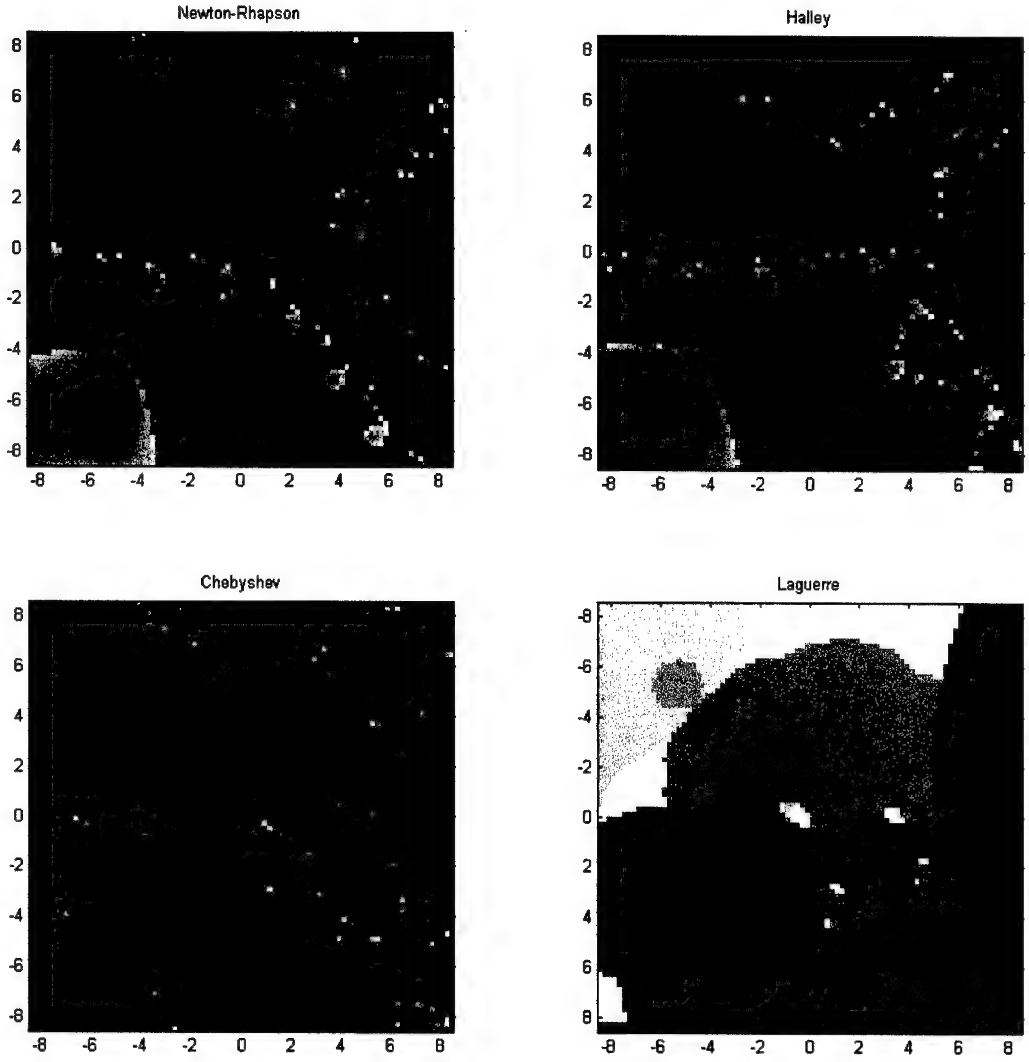
Figure 4.26. Mixed Roots – 5th Order



Roots

$$\left[9,\ i,\ -6-6i,\ -5+4i,\ -8i \right]$$

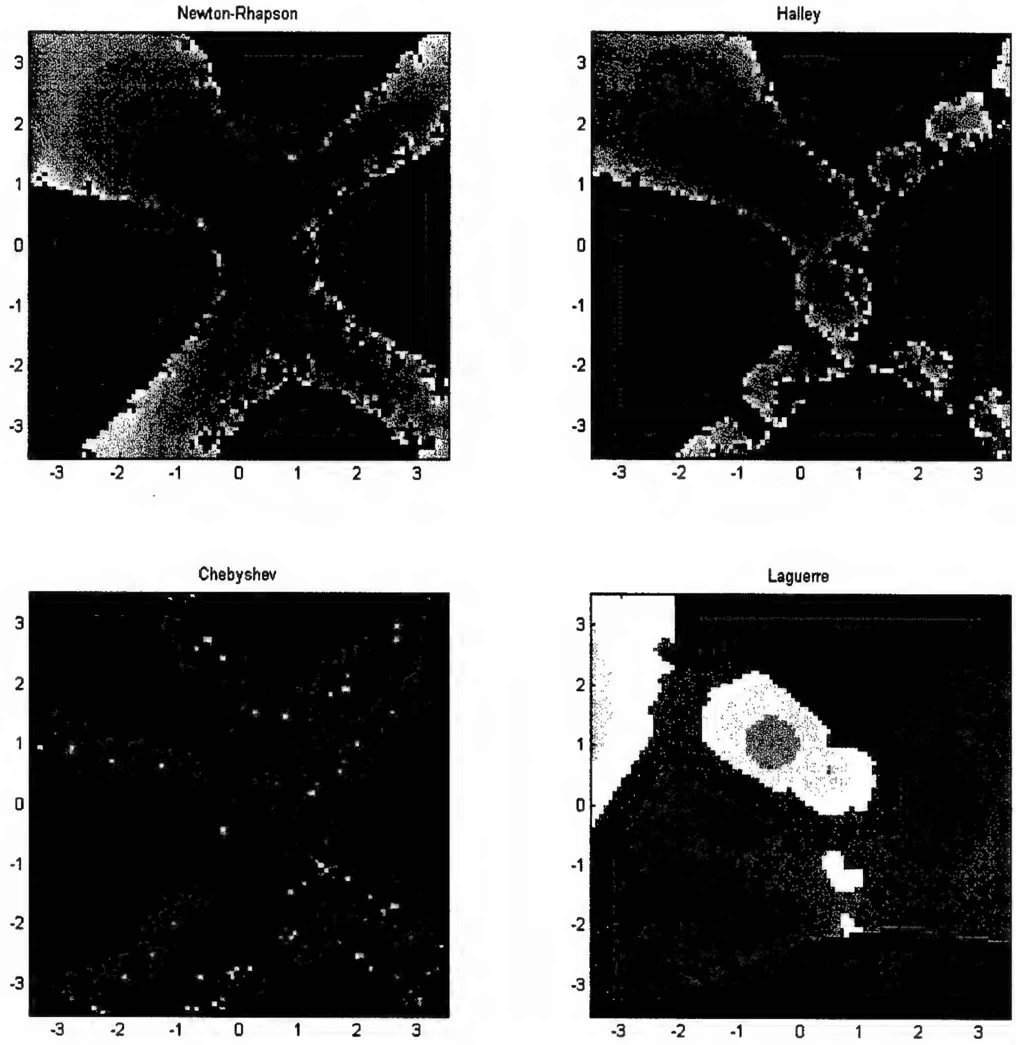
Figure 4.27. Mixed Roots – 5th Order



<u>Roots</u>

$$\left[8,\ 8i,\ -4-4i,\ -5-5i,\ -4+2i \right]$$

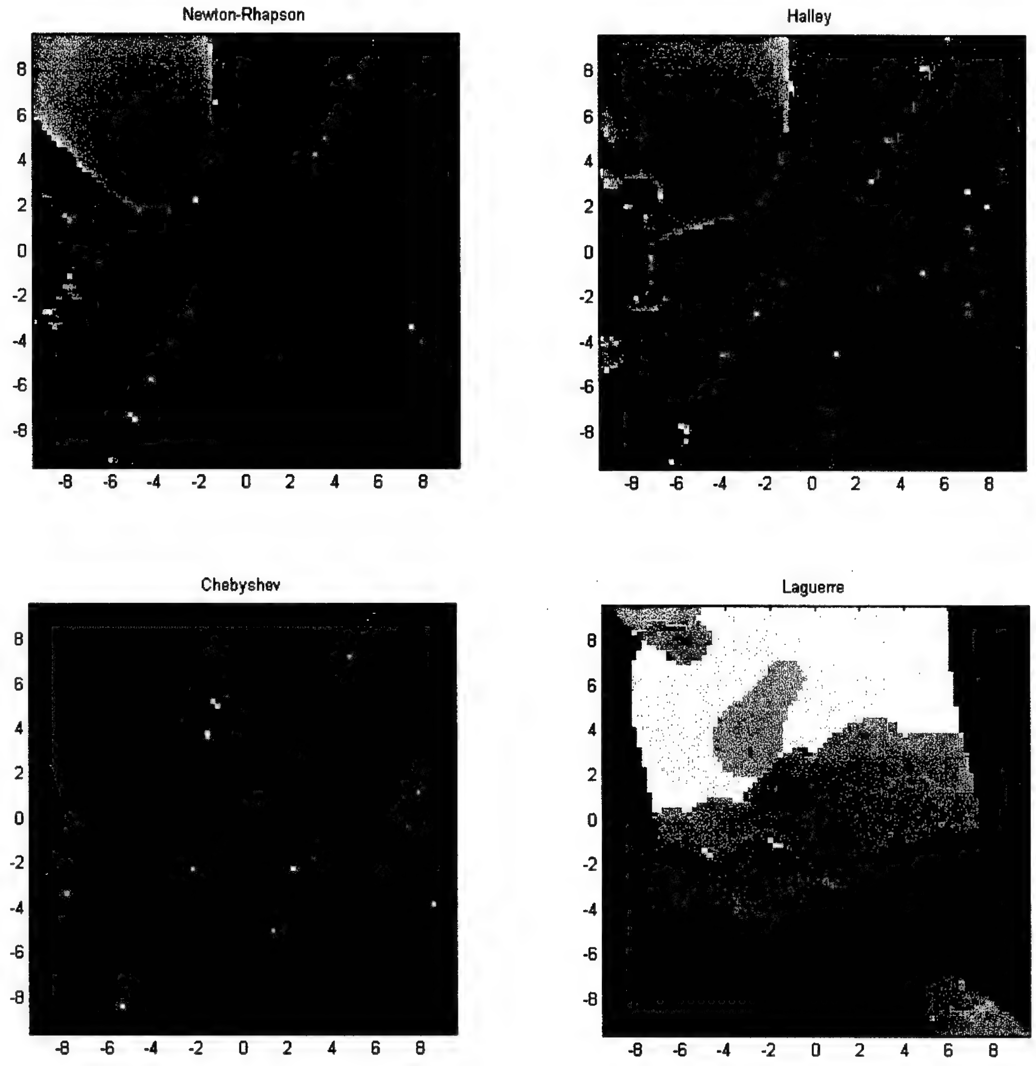
Figure 4.28. Mixed Roots – 5th Order



Roots

$$\left[-1,\ -\frac{1}{2}+i,\ -\frac{1}{2}+2i,\ 2,\ 1-3i\right]$$

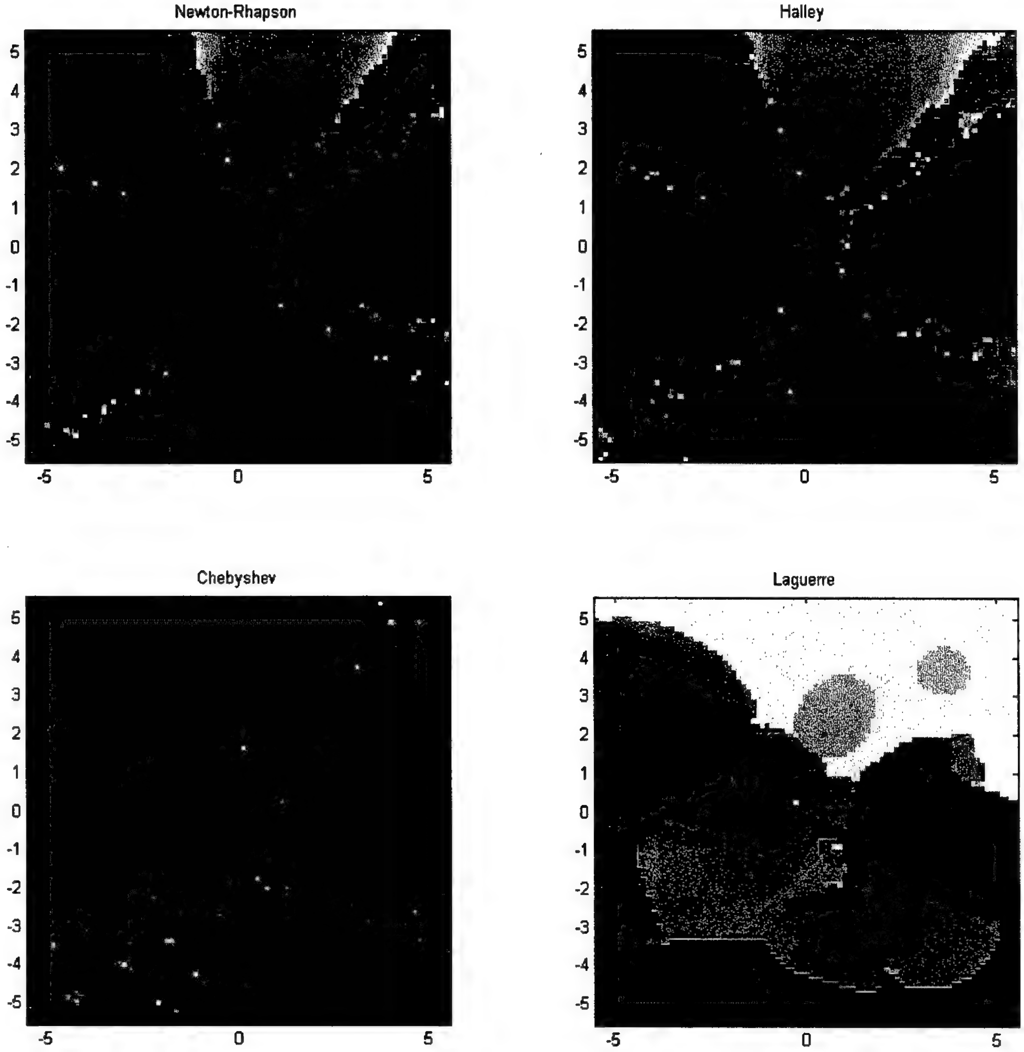
Figure 4.29. Mixed Roots – 5th Order



<u>Roots</u>

$$\left[-9,\ 9,\ -3+3i,\ -1+i,\ 1-i,\ 3-3i\right]$$

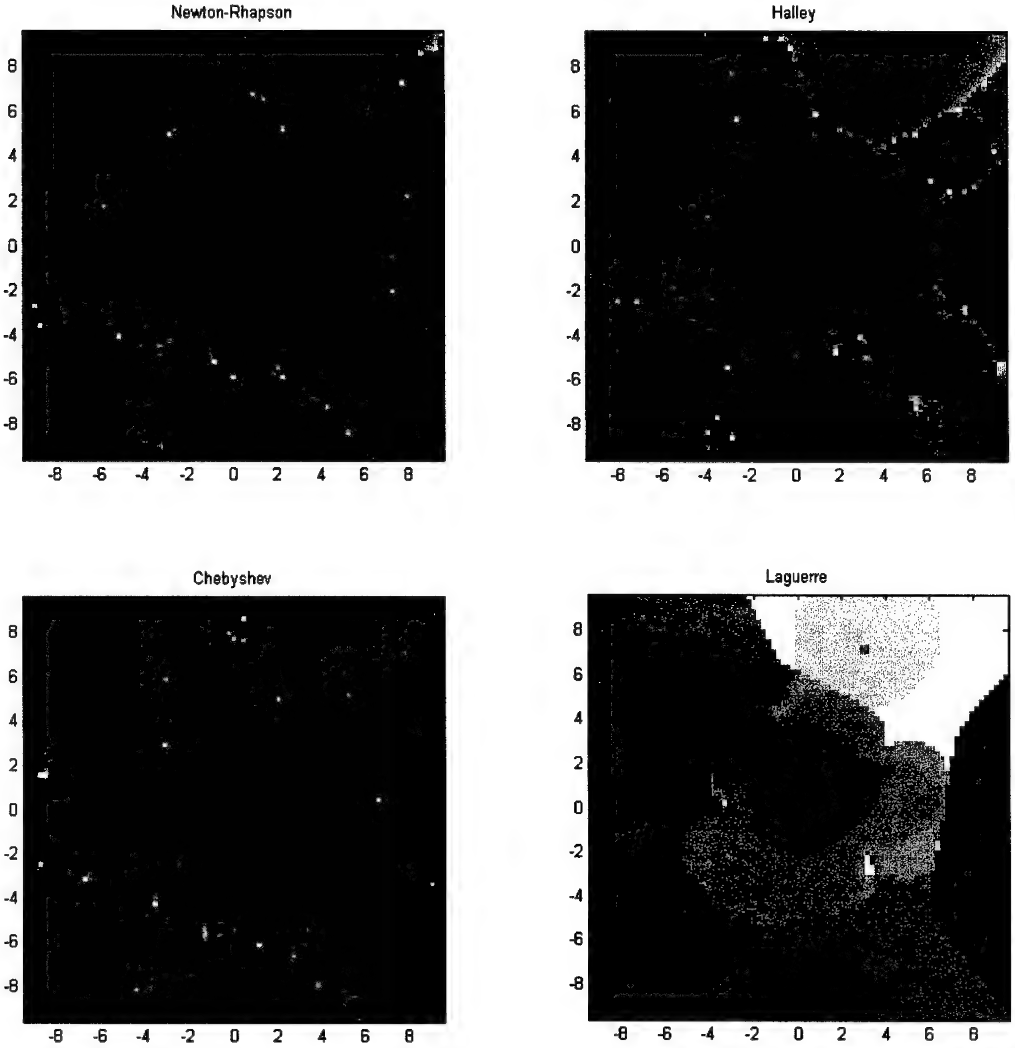
Figure 4.30. Mixed Roots – 6th Order



Roots

$$\left[-1,\ -\frac{1}{2}+i,\ -\frac{1}{2}+2i,\ 2,\ 1-3i,\ -5i\right]$$

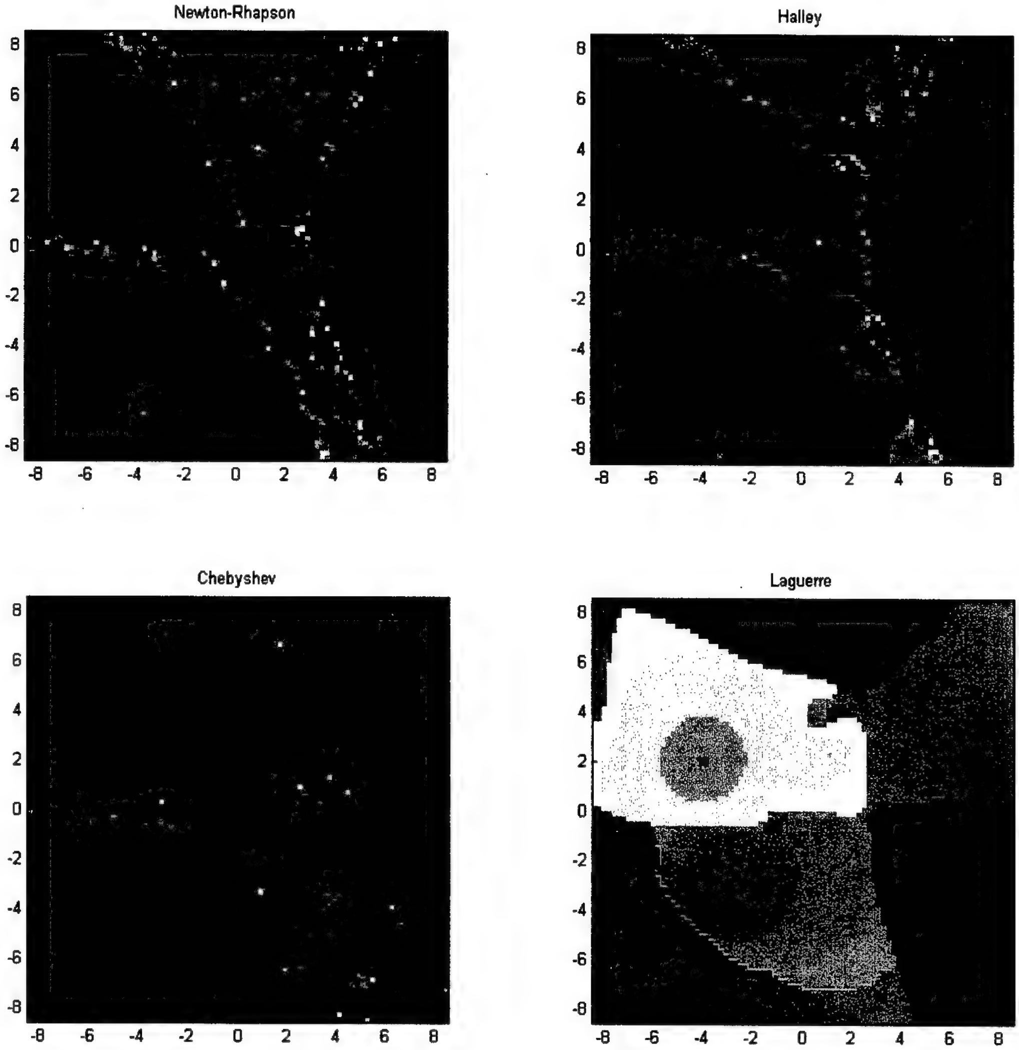
Figure 4.31. Mixed Roots – 6th Order



<u>Roots</u>

$$\left[9,\ i,\ -6-6i,\ -5+4i,\ -8i,\ 3+7i \right]$$

Figure 4.32. Mixed Roots – 6th Order



Roots

$$\left[8,\ 8i,\ -4-4i,\ -5-5i,\ -4+2i,\ 8+i \right]$$

Figure 4.33. Mixed Roots – 6th Order

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The Newton-Raphson, Chebyshev, Halley and Laguerre approximation methods, serve as powerful tools in evaluating complex polynomials' roots. These different methods, however, can yield different solutions from identical starting points. In determining any preference for the numerical methods, consideration must be given to the polynomial at hand, when do root finding methods converge and how long for convergence.

Whether low or high order, the Laguerre approximation method tends to fare better than other methods. With relatively simple basin boundaries, the method not only affords a greater effective radius of convergence but also increased behavior predictability. In many instances, the Newton-Raphson and Halley geometries are nearly indistinguishable for the same reason. When fixed points are a reflection of another, Halley's method assumes a much larger effective radius over the Newton-Raphson method, and it can even outdo Laguerre's method. Chebyshev's method, filled with complex boundaries and relatively small effective radii, remains the worst of the group.

With the methods relatively comparable in computational speed, the greater emphasis rests in basin shape, size, and complexity.

Ultimately though, the method of choice depends on the complex polynomial at hand.

B. RECOMMENDATIONS

While room for further research in this topic exists, a particular effort with respect to more numerical methods, calculating effective radii, and consideration for repeated roots would prove both challenging and rewarding.

LIST OF REFERENCES

Chapra, S.C. and Canale, R.P., *Numerical Methods for Engineers*, 2d Ed., McGraw-Hill Book Company, 1988.

Devaney, Rober L., *An Introduction to Chaotic Dynamical Systems*, 2d Ed., Addison-Wesley, 1989.

Hansen, E. and Patrick, M., "A Family of Root Finding Methods," *Numerische Mathematik*, v. 27, pp. 257-269, 10 February 1976.

Melman, A., "Geometry and Convergence of Euler's and Halley's Methods," *SIAM Review*, v. 39, pp. 728-735, December 1997.

Peitgen, H., Jurgens, H., and Saupe, D., *Chaos and Fractals: New Frontiers of Science*, Springer, 1992.

Pergler, Martin, "Newton's method and Newton basin fractals." [http://www.math.uchicago.edu/~pergler/genteach/newton/newton.html]. November 1999.

Popovski, D.B., "*A Family of One-Point Iteration Formulae for Finding Roots*," International Journal of Computation Mathematics Section B, v. 8 pp. 85-88, July 1979.

Press, W.H., and others, *Numerical Recipes in C: The Art of Scientific Computing*, pp. 279-280, Cambridge University Press, 1988.

Smith, Julius O., "The Fundamental Theorem of Algebra." [http://ccrma-www.stanford.edu/~jos/complex/Fundamental_Theorem_Algebra.html]. April 2001.

Strogatz, Steven H., *Nonlinear Dynamics and Chaos*, Perseus Books, 1994.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. BASIN CODE (MATLAB)

```
%******************************************************************%
% This MATLAB program computes basins of attractions for complex, analytic   %
% polynomials using various numerical methods. These methods include         %
% Newton-Rhapson, Chebyshev, Halley, and Laguerre.                           %
%                                                                            %
% User inputs: f: Analytic function:                                         %
%                   i.e. f=[1 0 0 1] ==> z^3 +1                              %
%              method1: Numerical Method                                     %
%                        i.e. 'N' = Newton-Rhapson, 'C' = Chebyshev          %
%                             'H' = Halley, 'L' = Laguerre                   %
%              t_limit: Maximum acceptable absolute difference between the   %
%                       computed and actual root for both axis.              %
%                        i.e. tol=.01+.01i                                   %
%              max_iteration: Maximum number of iterations before starting   %
%                             point becomes a member of the Julia set        %
%                              i.e. max_iteration=100                        %
%                                                                            %
% Notes:       With an nth degree polynomial generating n roots, the user must %
%              include n sets of the following codes to account for all roots. %
%                                                                            %
%                   if abs(p_n-actual_root(2)) <= tol                        %
%                        break                                               %
%                   end;                                                     %
%                                                                            %
%                   if abs(p_n-actual_root(1)) <= tol                        %
%                        root_color_code(real_counter,imag_counter)=1;       %
%                   end;                                                     %
%                                                                            %
%              Also, nth degree polynomial requires n+1 color assignments    %
%******************************************************************%

function basin_generator=basins(f,method1,t_limit,max_iteration)

% Defining/reseting initial conditions
iteration=0;
iteration_counter=0;
roots_found=0;
imag_counter=0;
real_counter=0;
n=length(f)-1;
method=char(method1);
tol=t_limit +t_limit*i;
```

```
% Determining the actual roots, coloring assignments, and complex plane
% boundaries and starting point step size
actual_root=roots(f);
root_colors=([1,0,0; 0,1,0; 0,0,1; 1,1,0; 1,0,1; 0,1,1; .5,1,0; 1,.5,0; .5,.5,.5; 1,1,1]);
bound=max([abs(max(imag(roots(f)))) abs(min(imag(roots(f)))) abs(max(real(roots(f))))
               abs(min(real(roots(f))))]);
imag_start_pt=-bound-.5;
imag_end_pt=bound+.5;
real_start_pt=-bound-.5;
real_end_pt=bound+.5;
step=bound/30

% Imaginary axis boundaries/do-loop
for imag_axis=imag_start_pt:step:imag_end_pt
    imag_counter=imag_counter+1
    real_counter=0;

    % Real axis boundaries/do-loop/assigning starting points
    for real_axis=real_start_pt:step:real_end_pt
      real_counter=real_counter+1;
      p_n_1=real_axis+imag_axis*i;

      % Resetting iteration counter/root 'a' measure for next starting point
      iteration=0;
      a=tol+1;

      % Iteration to determine convergence or Julia set member
      while iteration<=max_iteration

        % Fail safe -- No iteration necessary is starting on a root
        if polyval(polyder(f),p_n_1)==0
          break
        end

        % Newton-Rhapson Iterator
        if method=='N'
          p_n=p_n_1-polyval(f,p_n_1)/polyval(polyder(f),p_n_1);
        end

        % Chebyshev Iterator
        if method=='C'
          p_n=p_n_1-polyval(f,p_n_1)/polyval(polyder(f),p_n_1)-
                ((((polyval(f,p_n_1))^2)*(polyval(polyder(polyder(f)),p_n_1)))
                /(2*(polyval(polyder(f),p_n_1))^3));
```

```
end

% Halley Iterator
if method=='H'
  numerator1=2*polyval(f,p_n_1)*polyval(polyder(f),p_n_1);
  denominator1=2*(polyval(polyder(f),p_n_1))*(polyval(polyder(f),p_n_1))-
                (polyval(f,p_n_1))*polyval(polyder(polyder(f)),p_n_1);
  p_n=p_n_1-numerator1/denominator1;
end

% Laguerre Iterator
if method=='L'
  if iteration ~= 0
    p_n_1=p_n;
  else
    p_n_1=real_axis+(imag_axis)*i;
  end

  % Fail safe -- No iteration necessary is starting on a root
  if polyval(f,p_n_1)==0
    break
  elseif polyval(f,p_n_1)~=0
    G=(polyval(polyder(f),p_n_1)/polyval(f,p_n_1));
    H=G^2- polyval(polyder(polyder(f)),p_n_1)/(polyval(f,p_n_1));
    if (G + sqrt((n-1)*(n*H-G^2)))==0
      break
    end
    if (G - sqrt((n-1)*(n*H-G^2)))==0
      break
    end
    if abs(G + sqrt((n-1)*(n*H-G^2))) > abs(G - sqrt((n-1)*(n*H-G^2)))
      a=n/(G + sqrt((n-1)*(n*H-G^2)));
    else
      a=n/(G - sqrt((n-1)*(n*H-G^2)));
    end
  end
end

% Updating computed roots/iteration
iteration=iteration+1;
if method=='L'
  p_n=p_n_1-a;
else
  p_n_1=p_n;
end
```

```
  % Is computed root within tolerance of an actual root?
  if abs(p_n-actual_root(1)) <= tol
    break
  end
  if abs(p_n-actual_root(2)) <= tol
    break
  end
  if abs(p_n-actual_root(3)) <= tol
    break
  end

  % Extra statements for n roots
  % if abs(p_n-actual_root(4)) <= tol
  %   break
  % end
  % if abs(p_n-actual_root(5)) <= tol
  %   break
  % end
  % if abs(p_n-actual_root(6)) <= tol
  %   break
  % end
  % if abs(p_n-actual_root(7)) <= tol
  %   break
  % end
  % if abs(p_n-actual_root(8)) <= tol
  %   break
  % end
  % if abs(p_n-actual_root(9)) <= tol
  %   break
  % end

  % No root found, update variable counters for next iteration

end % while

% Iteration/Computed root trackers
iteration_counter(real_counter,imag_counter)=iteration;
computed_root(real_counter,imag_counter)=p_n;

% If computer root within tolerance of an actual root, do color assignment?
if abs(p_n-actual_root(1)) <= tol
    root_color_code(real_counter,imag_counter)=1;
  elseif abs(p_n-actual_root(2)) <= tol
    root_color_code(real_counter,imag_counter)=2;
```

```
        elseif abs(p_n-actual_root(3)) <= tol
          root_color_code(real_counter,imag_counter)=3;
        % Extra statements for n roots
        % elseif abs(p_n-actual_root(4)) <= tol
        %    root_color_code(real_counter,imag_counter)=4;
        % elseif abs(p_n-actual_root(5)) <= tol
        %    root_color_code(real_counter,imag_counter)=5;
        % elseif abs(p_n-actual_root(6)) <= tol
        %    root_color_code(real_counter,imag_counter)=6;
        % elseif abs(p_n-actual_root(7)) <= tol
        %    root_color_code(real_counter,imag_counter)=7;
        % elseif abs(p_n-actual_root(8)) <= tol
        %    root_color_code(real_counter,imag_counter)=8;
        % elseif abs(p_n-actual_root(9)) <= tol
        %    root_color_code(real_counter,imag_counter)=9;
        else
          root_color_code(real_counter,imag_counter)=10;
      end  % if

    end  % for real_axis
end  % for imag_axis

% Building the true color specification for root_color_code using
% an m-by-n-by-3 array of RGB values.
b(:,:,1) = iteration_counter;
b(:,:,2) = iteration_counter;
b(:,:,3) = iteration_counter;

% Scaling the colors to include iteration levels
for i = 1:length(root_color_code)
    for j = 1:length(root_color_code)
      b(j,i,:)= root_colors(root_color_code(i,j),:)*(((iteration_counter(i,j)
              /(max(max(iteration_counter)))))^.37);
    end
end

% Draw figure with appropriate title
figure(1)
image(b)
image(b,'XData',[-bound-.5 bound+.5],'YData',[-bound-.5 bound+.5]);
if method=='N'
  title('Newton-Rhapson');
end
if method=='C'
  title('Chebyshev');
```

```
end
if method=='H'
   title('Halley');
end
if method=='L'
   title('Laguerre');
end
axis square
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library...2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Department of Mathematical Sciences 1
   ATTN: Colonel Gary W. Krahn
   United States Military Academy
   West Point, NY 10996

4. Department of Mathematics..1
   ATTN: Chairman Michael A. Morgan
   Naval Postgraduate School
   1411 Cunningham Rd., Rm 341
   Monterey, CA 93943-5216

5. Department of Mathematics, Code MA/Ca 1
   ATTN: Professor David Canright
   Naval Postgraduate School
   1411 Cunningham Rd., Rm 341
   Monterey, CA 93943-5216

6. Department of Mathematical Sciences 1
   ATTN: Captain Bart D. Stewart
   United States Military Academy
   West Point, NY 10996